

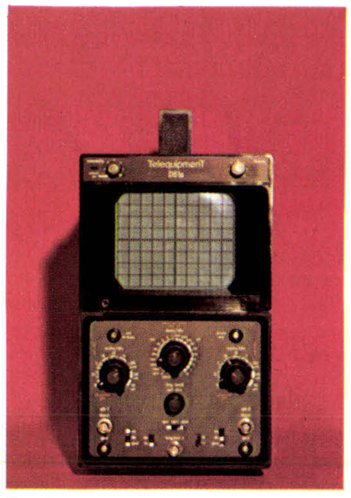
MICROPROCESSEURS / MICRO-ORDINATEURS / INFORMATIQUE APPLIQUÉE

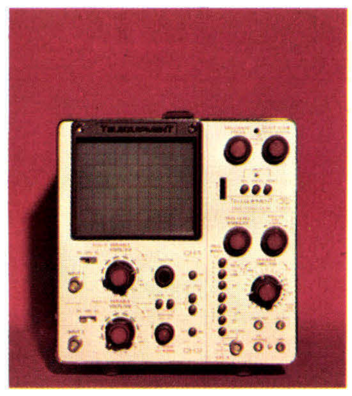
N° 2 Bimestriel – Novembre/Décembre 1978

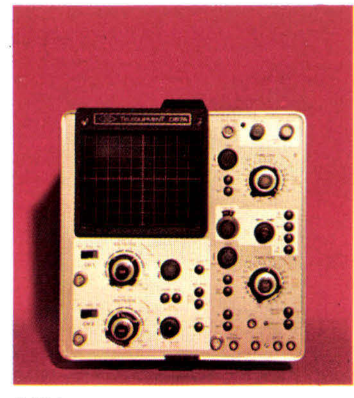
10F

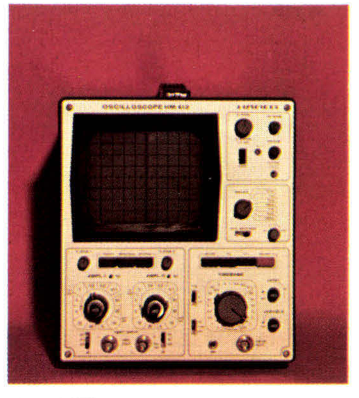
Suisse 4,50 Fs - Canada 2,50 $C - Italie 2.100 Lires - Espagne 130 Pes. - Algérie 10 Din. - Tunisie 1.000 Mil.

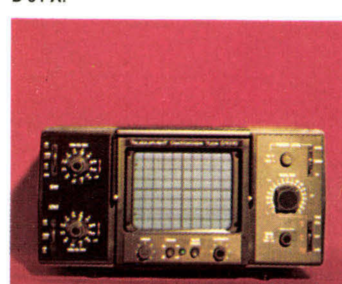
Chez PENTASONIC 6 mois de crédit gratuit sur les oscilloscopes.


D 61 A.


D 65.


D 67 A.


"HM 412/7"


D 1010 -

★★★★★★★★★

L'expédition de nos appareils n'est pas gratuite, mais :

- *Ils voyagent aux risques et périls de PENTASONIC.*
- *Ils ne sont pas expédiés par la poste, ni par la S.N.C.F., mais par un transporteur.*
- *Ils sont assurés. Si jamais un de nos appareils présente à l'arrivée (vérifiez avec le transporteur) le moindre défaut d'aspect, il vous sera immédiatement changé à nos frais.*

EMBALLAGE - TRANSPORT - ASSURANCE
En contre-remboursement, **78 F** -
Avec chèque à la commande, **53 F.**

★★★★★★★★★

TELEQUIPMENT

1700 F — **S 61 - 5 MHz**
Dimensions 28 x 16 x 37 cm. Tube 8 x 10 cm. Grande luminosité. Ampli vertical. bande pass. : 0 à 5 MHz.

2820 F — **D 61 A. Double trace 10 MHz**
Surface utile de l'écran : 8 x 10 cm. Bande passante : 10 MHz à 10 mV/cm.

4854 F — **D 65. Double trace 15 MHz**
Surface utile de l'écran : 8 x 10 cm. Bande passante : 15 MHz à 10 mV/cm.

7680 F — **D 67 A. Double trace 2 x 25 MHz**
1 mV/cm à 50 V/cm. Double base de temps. Balayage retardé. Déclenchement : normal, AC, DC, TV ligne et trame, automatique, HF-REJ.

NOUVELLE GAMME "D 1000"

2920 F — **D 1010 - Double trace 10 MHz**
5 mV à 20 V/div. Tension maxi 500 V. Balayage 0,2 s à 0,2µS/div., 40 ns en X5.

3231 F — **D 1011 - double trace 10 MHz**
version plus performante du 1010 mais caractéristiques principales identiques.

3880 F — **D 1015 - Double trace 10 MHz**
5 mV - 20 V/div. Tension maxi 500 V - Balayage 0,2 s à 0,2µS/div. 40 ns en X5.

4464 F — **D 1016 - Double trace 15 MHz**
version plus performante du 1015 mais caractéristiques principales identiques.

Les 4 appareils de cette nouvelle série sont conçus à partir d'une technologie avancée, garantie d'une grande simplicité d'emploi. Ils sont particulièrement adaptés à l'enseignement, au dépannage TV, à la fabrication et au contrôle industriel.

HAMEG

1445 F — **"HM 307"**
Simple trace. DC - 10 MHz (– 3 dB). Entrée à 12 possibilités ± 5 %. 5 mVcc - 20 Vcc/div.

2446 F — **"HM 312" Double trace 2 x 10 MHz**
Sensibilité 5 mV/cm à 20 V/cm. Déclenchement LPS - Tube 8 x 10 cm.

3269 F — **"HM 412/7" Double trace 2 x 15 MHz**
Tube 8 x 10 cm. AMPLIFICATEUR VERTICAL. Bande passante DC : à 15 MHz (– 3 dB), à 20 MHz (– 6 dB). Sensib. : 5 mVcc - 20 Vcc/cm. Balayage retardé.

5045 F — **"HM 512/7" Nouveau double trace 2 x 40 MHz**
2 canaux DC à 40 MHz, ligne à retard. Sensib. : 5 mVcc - 20 Vcc/cm. Régl. fin 1:3. Dim. de l'écran : 8 x 10 cm. Graticule lumineux.

SONDES OSCILLOSCOPES
Commutation X1 - X10 sur la sonde. Prix...... **192 F**

LEADER

3580 F — **"LBO 508". Double trace 2 x 20 MHz.**
10 mV/cm. Soustraction de trace XY. Base de temps 0,5µs à 200 mS/cm.

VOC

4234 F — **"VOC 5". Double trace 15 MHz**
Bandes passantes : DC de 0 à 15 MHz (– 3 dB), AC de 2 à 15 MHz (– 3 dB).

CREDIT IMMEDIAT

☎ 524-23-16
SUR LE PONT DE GRENELLE
5, rue Maurice-Bourdet - 75016 PARIS
A 50 mètres de la Maison de la Radio
Autobus : **70-72** (arrêt MAISON DE L'ORTF). MÉTRO : **Charles-Michels**

☎ 331-56-46
AUX GOBELINS
10 boulevard Arago - 75013 PARIS
Métro : Gobelins

Ouvert tous les jours (sauf dimanche) de 9 h à 12 h 30 et 14 h à 19 h 30

Sommaire

MICRO SYSTEMES

Pages





Notre couverture :

Un microprocesseur bipolaire rapide de 4 bits, le SF.C 92901 (cliché Thomson-CSF, Jacques Pierre).

Une nouvelle génération de circuits : les CODECs. Le DF 331/332 (Loi μ) et le DF 341/342 (Loi A) réalisent l'interface entre le monde analogique et le système logique dans toute transmission P.C.M. (Doc. Siliconix).

Président-directeur général :
Directeur de la publication :
Jean-Pierre Ventillard

Rédacteur en chef :
Alain Tailliar

Comité de rédaction :
Jean-Michel Cour — André Doris
Jean Frémaux — Jean-José Wanègue.

Rédaction :
15, rue de la Paix, 75002 Paris
Tél. : 296.46.97

Maquette : Josiane Garnier.

Publicité :
S.P.E.
Tél. : 296.46.97

Abonnements :
2 à 12, rue de Bellevue
75940 Paris Cedex 19
Tél. : 200.33.05
1 an (6 numéros) : 45 F (France)
70 F (Etranger)

Société Parisienne d'Edition
Société anonyme au capital de 1 950 000 F
Siège social : 43, rue de Dunkerque, 75010 Paris
Direction - Administration - Ventes :
2 à 12, rue de Bellevue, 75940 Paris Cedex 19
Tél. : 200.33.05 - Télex PGV 230472 F

Dépôt légal 3e trimestre 78
N° éditeur : 644
Distribué par SAEM Transports Presse
Imprimerie : La Haye-Mureaux

TOUS RENSEIGNEMENTS MICROPROCESSEURS
RIVE DROITE 524.23.16
RIVE GAUCHE 331.56.46
OU SUR PLACE

PENTA-MAGAZINE

12-78

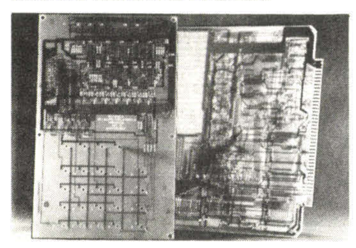
Editorial

Nous sommes anti-trust !

PENTASONIC est une association de techniciens. Nous avons ouvert un second point de vente pour le rester. Dans chaque magasin, vous trouverez une équipe légère travaillant selon ses propres méthodes, car nous sommes certains de vous apporter un meilleur service en prenant des voies concurrentes.



Le "NUMÉRO 1" ou l'anti-gadget : MKII MOTOROLA



LE fait de spécifier qu'il est à la base du 6800 MOTOROLA suffit à le considérer comme l'un des systèmes les plus souples, les plus performants, mais surtout, l'un des plus faciles à utiliser.

Il existe des microprocesseurs plus puissants mais, à notre avis, aucun de plus sympathique. Allié à sa facilité d'emploi (manuel d'utilisation en français), vous bénéficiez d'un service après-vente digne de MOTOROLA.

La vocation du MK II, outre l'initiation, est d'être le premier maillon d'un système puissant capable de gérer télétype, visu et floppy; d'être extensible grâce à ses cartes interfaces... qui existent!

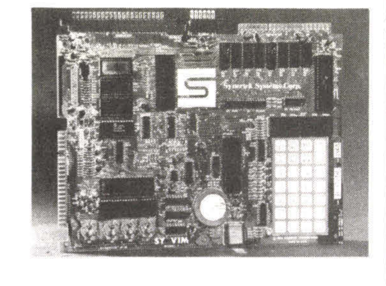
Le MK II c'est l'anti-gadget des systèmes de développement.

1968 F. Avec notice d'utilisation en français.

L'OUTSIDER : le VIM I SYNERTEK

Nous considérons le 6502 comme un microprocesseur très légèrement moins puissant que le 6800, bien que très proche de celui-ci. Le système de développement qui lui est associé est, par contre, très étoffé. Ses 4 K de moniteur d'origine, son 1 K de RAM, sa gestion directe de télétype ou de visu, en font un système complet. La plupart des extensions peuvent se monter sur la carte et particulièrement 3 K de RAM supplémentaires, ainsi que 32 K de ROM. Parmi ces 32 K de ROM on pourra implanter un gros BASIC de 8 K, dont Synertek annonce la commercialisation. Autre avantage le VIM I se vend tout monté.

2350 F avec notice d'utilisation en français.

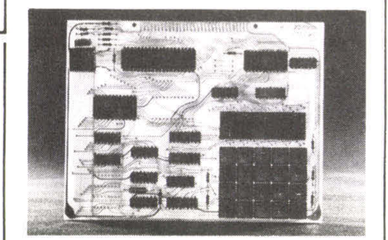


FAITES-LE DONC VOUS-MÊME!

Voulez-vous construire votre système 6800 vous-même? Nous tenons, dans ce cas, à votre disposition un classeur comprenant le circuit imprimé, les plans, et les notices d'un système de développement. Une fois monté, il gérera une visu ou un télétype. Il disposera d'1 K de mémoire, un PIA, un ACIA et un moniteur MIK-BUG. Ce kit, destiné aux amateurs avertis, bénéficie de l'assistance technique PENTASONIC. MOTOROLA le commercialise, dans sa version d'origine sous l'appellation MKI.

Le circuit imprimé 200 F.

Un petit malin qui cache son jeu



Ce produit à base de SCMP détient certainement l'un des meilleurs rapports prix/performances du marché. Particulièrement destiné aux contrôles de processus, c'est le système d'initiation par excellence. Toutes les cartes d'extension, ainsi que le produit lui-même sont fabriqués en France.

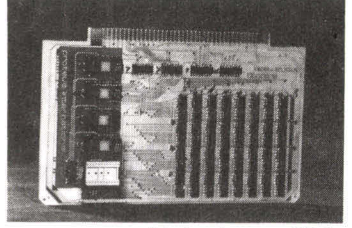
Série UC 1000 EMR, 985 F

LA MEILLEURE ET LA PIRE DES CHOSES

Le soft est la meilleure et la pire des choses. La meilleure lorsqu'il s'agit d'aider l'utilisateur par une plus grande souplesse de langage, une plus grande puissance. La pire lorsqu'on tombe dans la facilité et qu'on l'utilise mal : par exemple gérer un clavier ou une visu directement. Le soft n'est pas le Bon Dieu! Il ne peut pas tout faire et s'il s'occupe à servir ses propres intérêts, il ne lui restera que peu de temps à vous consacrer.

Le but, notre but, n'est pas d'afficher des petits caractères sur un printer ou sur une télévision, mais, de travailler en concordance avec notre système. N'oubliez jamais que le soft n'est que le prolongement de vous-même : avec vos qualités et ses défauts. Du soft... PENTABUG ! qui se présente sous la forme de 1 MM 2716 se montant directement sur les emplacements EPROM du MK II et permet de brancher directement nos cartes de visualisation et nos claviers sur les KITS MAZEL et MK II • Mis au point par M. DAUTREVAUX..................... 195 F TTC Comprenant 1 MM 2716 + 1 notice + 1 listing

DO YOU SPEAK BASIC?



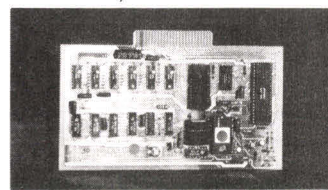
Cette carte, fabriquée par PROTEUS INTERNATIONAL vous permet de compléter votre système informatique et d'accéder à un "personal computer" de haut de gamme. Il existe différents types de Basic. Le Tiny-Basic (de 2 à 4 K), qui se rapproche plus du gadget que de l'outil de travail et qui permet entre autre d'apprendre le langage. Le Full-Basic (de 4 à 7 K) qui est le langage d'origine et l'Extended-Basic que l'on considère comme le plus souple, et l'un des plus puissants. Grâce à celui-ci, vous aurez accès à la fantastique bibliothèque existante : calcul scientifique, gestion de stock, aterrissage de fusée, jeux, etc.

Carte basic 1820 F

- Ce moniteur comprend et exécute 28 fonctions.
- Il permet en outre d'adresser un ACIA en 8010-8011 et de commander tout interface travaillant en ASC II.

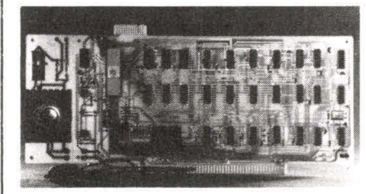
DEUX SYSTÈMES DE VISUALISATION D'INFORMATIONS SUR TERMINAL VIDÉO

Une carte pour voir, à la française...



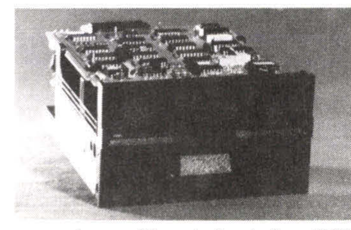
Cette carte est un outil de travail professionnel. L'entrée des informations se fait en ASCII, en série ou en parallèle. La mémoire de l'écran se trouve sur la carte, ainsi que la gestion de celui-ci. La sortie est en vidéo directe, c'est-à-dire qu'elle se branche directement sur l'entrée vidéo de votre téléviseur, ou, sur l'entrée UHF par l'intermédiaire d'un modulateur. Elle gère 16 lignes de 64 caractères en vitesse réglable jusqu'à 12.000 bauds en série.
Carte interface vidéo SFF 96364. 1512 F en kit, avec notice en français.

Une carte pour voir à l'américaine...



Cette carte a été conçue par MOSTEK aux U.S. et modifiée pour fonctionner sur le standard français. Elle est commercialisée montée et testée. Elle possède un avantage évident pour les radio-amateurs : elle fonctionne en Baudot à 47,5 bauds et peut faire la conversion ASCII/Baudot. L'entrée des informations se pratique en série ou en parallèle, de plus, le curseur est adressable en absolu ou en relatif ce qui permet un pseudo-graphisme.
Carte interface vidéo MK 3870, 1580 F montée.

MINI FLOPPY DISK-DRIVE ou une mémoire d'éléphant



Dernier maillon de la chaîne 6800 MKII : l'extension "FLOPPY"! De la même façon que ses grands frères ordinateurs, le 6800 peut avoir besoin d'une mémoire de masse, non volatile et d'accès rapide. Cette nouvelle technologie, le Floppy, entre dans l'univers du Hobbyste.

Ce lecteur offre la possibilité de stocker 2,6 mégabits de mémoire sur un disque souple avec un temps d'accès moyen aux information de 500 mS. Il peut être livré indifféremment de marque SHUGART ou PERTEC. Il s'adapte également sur APPLE. La carte de formatage pour 6800 sera disponible fin novembre et compatible avec tous les systèmes 6800.
Le lecteur double densité 2100 F, le Floppy Disk 49 F.

Quand la force de frappe devient caresse...

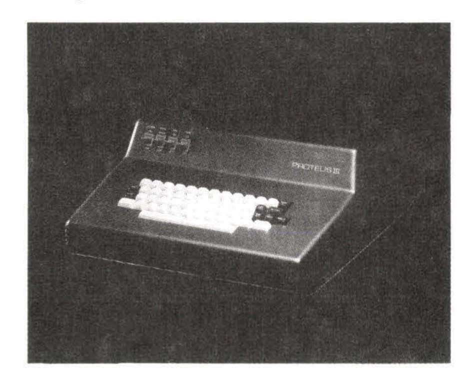


Nous vous présentons un clavier aussi moderne que votre microprocesseur. Malgré la technique employée (pas de contacts) et la sophistication de sa gestion, il est vendu à un prix très compétitif. Il devient très vite lassant, en cours de programmation de taper sur un clavier qui rebondit ou dont les touches manquent de souplesse. Sa sortie s'effectue sur 7 bits + strobe et il s'alimente uniquement en 5 volts. Il est géré par un microprocesseur GI spécialisé pour ce travail.
Clavier ASCII, dernier-né de KEY-TRONIC© 1180 F.

Carte
BUS EXORCISER
vendue par le club 6800
149 f sans connecteur

IL SAVAIT TOUTES LES CHOSES ET PARLAIT TOUTES LES LANGUES...

PROTEUS : dieu grec, qu'on appelait le sage d'Egypte, car il vivait dans une petite île au large des côtes d'Alexandrie.
Proteus savait toutes les choses et parlait toutes les langues. A ce titre, il était très sollicité, les gens se pressaient pour l'interroger. Mais comme Proteus savait, en outre, prendre toutes les formes, lorsqu'il était excédé, il devenait arbre, poisson ou fleur.
Un jeune homme réussit cependant, à force de gentillesse et de patience, à séduire Proteus, qui lui livra tous ses secrets...



GÉRER POUR RÉUSSIR.

Proteus III, système complet de micro informatique de gestion, est maintenant là pour vous seconder fidèlement en traitant rapidement, efficacement, clairement tous les problèmes de classement, de comparaison, recherche, identification, que vous étiez seul jusqu'ici à pouvoir résoudre pour gérer votre activité.
Or, que vous soyez Chef d'Entreprise, Expert comptable, Médecin, Dentiste, Pharmacien, Avocat, Gérant d'immeubles, Hôtelier, Agent de voyages, Ingénieur ou Architecte, ... vous avez tous un point commun : vous savez que pour réussir, il faut gérer mieux.

PROTEUS III UNITÉ CENTRALE

25 K de mémoire ou 41 K.
16 K-Ram (Réf. Proteus III-A).
32 K-Ram (Réf. Proteus III-B).
1 K-Mémoire de page écran.
8 K-Rom (Basic).
Unité centrale 6800 travaillant à 0,894 MHz.
Clavier, 53 touches capacitives.
128 caractères dont ASCII majuscules + caractères graphiques.
Sortie UHF et vidéo.
Interface imprimante série : 20 mA, TTY, RS 232, gérée par le Basic.
Basic type étendu, capable de gérer 3 périphériques simultanément (clavier-écran vidéo, imprimante TTY, magnéto K7), plus, éventuellement 1 à 3 floppy disk.
Magnétophone à K7 travaillant à 300 Bauds, utilisé comme mémoire de masse.
Prix .... 10.758 F

PROTEUS MONITOR

Moniteur vidéo de 30 cm, haute résolution, avec écran standard Proteus (européen).
Prix .... 1.680 F

PROTEUS PRINT

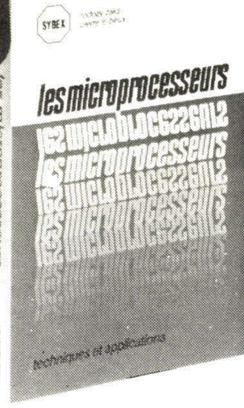
Imprimante sur papier normal (non métallisé), travaille sur 80 colonnes à 1.200 Bauds (120 ch./s).
Prix .... 10.240 F

PROTEUS FLOPPY

(livraison fin novembre).
Mini-Floppy travaillant en double densité, double face.
2,6 Megabits par disquette.
Vitesse de transfert de 250 K-Bauds.
Temps d'accès moyen à un fichier : 500 ms.
Prix .... 4.704 F

MICRO-ELECTRONIQUE — MICRO-INFORMATIQUE

BEST-SELLER en FRANCE et aux U.S.A.



LES MICROPROCESSEURS

TECHNIQUES ET APPLICATIONS

RODNAY ZAKS ET PIERRE LE BEUX

L'ouvrage de base sur les microprocesseurs pour toute personne ayant une formation technique ou scientifique. Il s'agit d'un livre conçu pour la formation, qui se lit facilement, malgré sa technicité. Il enseigne pas à pas tous les concepts et techniques liés aux microprocesseurs, depuis les principes de base jusqu'à la programmation. Indépendant de tout constructeur, il présente les techniques « standards » valables pour tout microprocesseur, y compris l'interconnexion d'un système « standard ». Il introduit le MPU, son fonctionnement interne, les composants d'un système (ROM, RAM, UART, PIO, autres), leur interconnexion, les applications, la programmation et les problèmes liés au développement d'un système.
Un ouvrage de 320 pages, format 16 x 24.
200 illustrations. Prix : 95 F.

INFORMATIQUE

D.J. DAVID

Cours d'initiation à l'informatique (ENS). Langages de programmation : Fortran, APL. Fonctionnement interne des ordinateurs, l'esprit informatique, modèles schématiques des applications, cartes-contrôle ; IBM, CDC, UNIVAC, CII et Philips. 336 pages.

NIVEAU 3 — *PRIX 65 F*

LOGIQUE INFORMATIQUE

Marc FERRETTI

Cours et exercices sur la théorie des ensembles et l'algèbre de Boole. 160 pages.

NIVEAU 3 — *PRIX 25 F*

TECHNIQUE POCHE Nº 4

INITIATION A LA MICRO-INFORMATIQUE : LE MICROPROCESSEUR

136 pages — *MELUSSON*
NIVEAU 2 — *PRIX 27 F*

ETSF ÉDITIONS TECHNIQUES ET SCIENTIFIQUES FRANÇAISES

ELEMENTS ESSENTIELS DE L'ELECTRONIQUE ET DES CALCULS DIGITAUX

D. ULRICH

Logique électronique. Logique informatique. Calculateurs à circuits logiques. Réalisation des calculateurs. Le transistor en commutation. Multivibrateurs. Montages logiques de base. Fonctions logiques. Algèbre de Boole. Calculs binaires. 304 pages.

NIVEAU INGÉNIEUR — *PRIX 86 F*

LE HARDSOFT

ou la pratique des microprocesseurs

M. OUAKNINE et R. POUSSIN

Principes généraux. Fonctionnement et jeu d'instruction d'un système construit autour du microprocesseur 8080 A. Trois applications réelles avec schémas et programmes : le lecteur pourra ainsi réaliser lui-même son système d'initiation. 200 pages.

NIVEAU 3 — *PRIX 56 F*



MICRO-INFORMATIQUE MICRO-ELECTRONIQUE

HENRI LILEN ET PIERRE MORVAN

DICTIONNAIRE DE DEFINITIONS AVEC LEXIQUE ANGLAIS-FRANÇAIS



PRIX 92 F

Le dictionnaire que vous attendiez en français



S'inspirant de l'ouvrage « MICROCOMPUTERS and MICROELECTRONICS Dictionary and Guide » mais tenant compte des acquis de la langue française et de la législation, ce dictionnaire en français regroupe environ un millier de mots, sigles et expressions utilisés en micro-électronique et en micro-informatique, avec leurs définitions françaises et leur traduction (français-anglais).
Il comporte également un lexique de rappel anglais-français.
Nombreux sont les mots, sigles ou expressions qui y figurent et que l'on ne trouvera dans aucun autre dictionnaire publié à ce jour.
352 pages. Format 13,5 x 21.

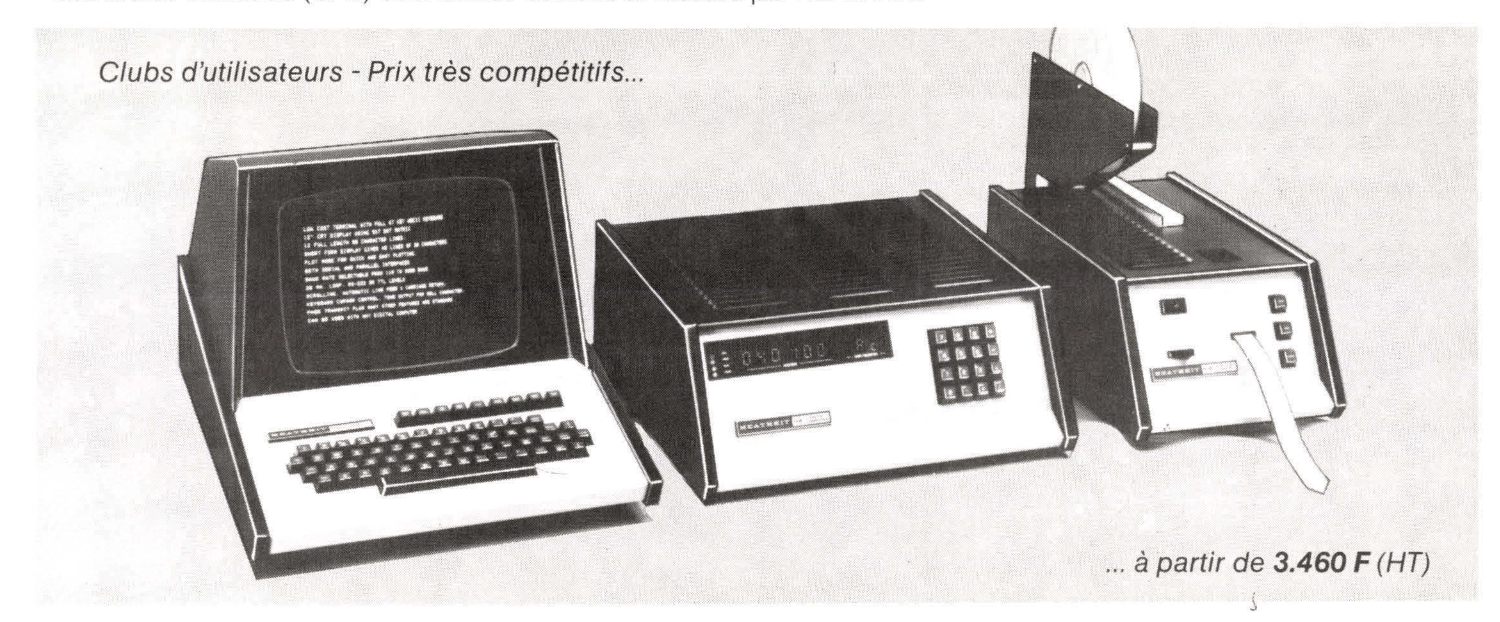
En vente chez votre libraire habituel ou à la
LIBRAIRIE PARISIENNE DE LA RADIO
43, rue de Dunkerque, 75480 PARIS CEDEX 10
Conditions de vente par correspondance : jusqu'à 100 F, 15 % de la commande plus 3 F Rdé, facultatif. Au-dessus de 100 F : taxe fixe 18 F Rdé obligatoire.

l'informatique à votre portée
avec les ordinateurs "HEATHKIT"

... deux systèmes complets, étudiés spécialement pour vous, techniciens, scientifiques, éducateurs, amateurs avertis, responsables d'entreprises.

- Système H8, mots de 8 bits, avec microprocesseur 8080 A et capacité de mémoire centrale extensible à 32 K mots (RAM statique).
- Système H11, mots de 16 bits, avec microprocesseur LSI 11, et capacité de mémoire centrale extensible à 32 K mots (RAM - MOS statique).
- Nombreux périphériques : Console de visualisation - Perforateur/lecteur de bande - Imprimante DEC LA 36 (30 cps) - Mémoire masse sur cassettes, disques souples - Interfaces I/0 séries, parallèles. Ce matériel est livré en KIT* ou en ordre de marche, avec logiciel très complet, comprenant même le Basic 8K (en option : Basic 12 K et gestion fichier) vous permettant de développer vos applications à l'infini.

* Les unités centrales (CPU) sont livrées câblées et testées par HEATHKIT.





KIT MICRO-ORDINATEUR d'initiation "6800"
Moniteur 1 K ROM - 256 x 8 RAM - Extension à 512 x 8 RAM - Affichage 6 digits, 7 segments - Clavier hexadécimal - Possibilité d'extension cassette, etc... - Livré avec alimentation stabilisée - Manuel en Anglais.

Prix : **1.990 F** (TTC) + port 20 F.

COURS sur les MICROPROCESSEURS
Le plus complet à ce jour, 8 chapitres, 800 pages, rédigé en Anglais, assorti d'exemples et expériences, à l'aide de 62 composants électroniques (y inclus).

Prix : **820 F** (TTC) + port 10 F.

Bon à découper, à adresser à :

FRANCE : Heathkit, 47 rue de la Colonie, 75013 PARIS, tél. 588.25.81
BELGIQUE : Heathkit, 16 Av. du Globe, 11.90 BRUXELLES, tél. 344.27.32

Je désire recevoir votre catalogue 16 pages couleurs, en Anglais, contenant tous renseignements sur : (cocher la case)

☐ Ordinateurs H 8 et H 11 ☐ Cours sur les microprocesseurs

Je joins 4 Francs en timbres, pour frais d'envoi

Nom, prénom ____________________

N° ________ Rue ____________________

Code postal ________ Ville ____________________

M.S. 11-78

Centres de démonstration :

PARIS (6e), 84 Bd Saint-Michel, tél. 326.18.91
LYON (3e), 204 rue Vendôme, tél. (78) 62.03.13

Editorial

Devant l'élan spontané avec lequel chacun d'entre vous est venu participer au succès de la première parution de « Micro-Systèmes », nous sommes maintenant convaincus, tout comme vous que sa naissance se justifiait.

Un volumineux courrier ainsi que de nombreux appels téléphoniques témoignent d'une remarquable réussite pour ce lancement qui n'a pas manqué d'être salué par la presse écrite et la télévision et pour lequel nous remercions tous ceux qui parmi vous nous ont si aimablement adressé leurs vœux et leurs éloges.

En ayant dès maintenant choisi de devenir lecteur de **« Micro-Systèmes »,** vous venez de constituer un groupe qui, grâce à la révolution technologique qu'il anime, verra tomber le privilège qui faisait trop souvent de l'informatique l'exclusive propriété des grands organismes ou des fonctions administratives de nos grandes entreprises.

L'informatique doit se démocratiser. A présent la technologie le permet, il ne manquait plus que vous.

Ainsi, que vous soyez professionnel, étudiant ou amateur d'électronique, que vos motivations soient lucratives ou récréatives, vous aiderez à la diffusion d'une technique moderne qui demain sera le support indispensable de bien des éléments de notre environnement quotidien.

Devant de telles ambitions, **« Micro-Systèmes »** ne peut pas se contenter d'être une simple revue qui se laisse confortablement lire et vous abandonne ensuite à la dernière page.

Il s'agit, en plus de nos articles, de proposer dans ce journal un ensemble de services qui mis à la disposition des lecteurs leur offre le moyen de pénétrer sur le terrain du vécu et de valoriser ainsi leurs efforts d'apprentissage en concrétisant et en stimulant leur savoir par des réalisations pratiques entre autres possibilités.

C'est la raison pour laquelle, vous pouvez si vous le souhaitez, commencer dès ce numéro la réalisation de votre « Ordinateur Personnel ». A cet effet nous avons demandé au groupe **Proteus-International** d'étudier pour vous un micro-ordinateur simple d'emploi et accessible aux non-spécialistes de l'informatique. De par ces caractéristiques il se situe dans la classe des « Personal-Computers » avec ses 16 k de mémoire.

Pour être utilisable par tout le monde ce système se doit d'utiliser un langage simple. Si un microprocesseur a son propre langage, ô combien hermétique, il est possible, malgré tout, de dialoguer avec lui en lui imposant un mode de conversation encore assez proche de celui que nous utilisons entre nous. Pour ce faire on intercale entre lui et l'opérateur un véritable interprète formé suivant nos convenances.

En conséquence la solution retenue pour « Micro-Systèmes I » est une programmation en langage **Basic étendu,** choisi pour sa facilité de compréhension et d'utilisation.

Maintenant il ne vous reste plus qu'à prendre le départ de cette merveilleuse expérience qui d'un seul coup va vous faire découvrir un monde nouveau pour lequel vous aurez choisi la position la plus enviable, à savoir : être celui qui sait. Ainsi au travers de cette revue et grâce aux initiatives inventives qu'elle aura su provoquer chez vous, vous constituerez une famille de pionniers qui rapidement sera la source d'applications inédites, à l'instar des « hobbyists » Américains.

Toute l'équipe de **« Micro-Systèmes »** vous remercie de l'enthousiasme que vous lui avez manifesté et espère bientôt vous rencontrer par l'intermédiaire de votre courrier qui sera toujours le bienvenu.

Jean-José WANEGUE

Les micro-ordinateurs individuels : mythe ou réalité ?



Photo 1. – Un système individuel type comme celui représenté ci-dessus comprend : un boîtier qui inclut le micro-ordinateur et le clavier, un poste de télévision domestique et un simple magnétophone à cassettes pour le stockage des programmes (Doc. Sybex).

Les micro-ordinateurs à usage personnel sont nés à la suite d'un accident journalistique. Depuis l'introduction des premiers microprocesseurs fin 1971, les micro-ordinateurs avaient été construits et assemblés à des fins industrielles. Fin 1974, l'éditeur de « Popular Electronics » (l'une des principales revues de vulgarisation électronique aux Etats-Unis) Leslie Solomon, de passage à Albaquerque, au nouveau Mexique, visite la firme Mits. Dans le coffre de la voiture du président de cette petite société se trouve un micro-ordinateur et Leslie Solomon propose de le baptiser « Micro-ordinateur Personnel ». Il importe ensuite de trouver un nom pour ce micro-ordinateur et c'est la petite fille du rédacteur en chef qui propose le nom de « Altair » tiré de l'un des épisodes de la série « Startrek » qui passe en ce moment à la télévision.

Dans le numéro de janvier 1975 le Altair est décrit comme le premier micro-ordinateur à vocation personnelle. Le succès est foudroyant, les commandes affluent par dizaines puis par centaines et la petite firme Mits devient rapidement le premier constructeur de micro-ordinateurs au monde. Pendant longtemps, Mits détiendra plus de la moitié du marché. La demande ne cesse d'augmenter et de nombreuses firmes se lancent les unes après les autres à la poursuite de ce nouveau marché, c'est ainsi qu'apparaissent le IMSAI 8080, puis tous les autres micro-ordinateurs connus aujourd'hui. L'un des résultats surprenants du succès inattendu du Altair est la naissance d'une norme nouvelle pour le bus interne des micro-ordinateurs. C'est le bus désormais connu sous le nom de bus « S 100 » et adopté depuis par la majorité de tous les systèmes.

Que pouvaient faire ces premiers micro-ordinateurs individuels ?

A proprement parler, rien ! Leur attrait principal était d'offrir la puissance de calcul d'un ordinateur à très faible coût (quelques milliers de francs). Les premières années (1975-1976) furent les années dites des « Hobbyists », c'est-à-dire les années où presque tous les systèmes à micro-ordinateurs furent achetés par des bricoleurs dans la tradition des amateurs radio. Ces amateurs étaient prêts à accepter les longs délais de livraison, les nombreux problèmes de fiabilité et le manque quasi total de logiciel pour ces premiers micro-ordinateurs. La plupart de ces systèmes furent utilisés pour créer des jeux sophistiqués mais surtout comme passe-temps.

La deuxième étape importante

Encore aujourd'hui, le logiciel disponible pour micro-ordinateur est très réduit.

fut marquée par une exposition dite le « Personal computing show » organisée à Atlantic City dans le New Jersey aux Etats-Unis en août 1976 par John Dilks. Cette exposition, qui ne devait réunir qu'un assez faible nombre de participants, en attira plus de sept mille. Le mouvement était révélé alors au monde extérieur. C'est à cette date que furent annoncées les dizaines d'expositions de micro-ordinateurs qui, depuis, furent organisées à travers tous les Etats-Unis. Cette exposition devait révéler l'existence d'un marché nouveau de dizaines de milliers puis de centaines de milliers d'utilisateurs potentiels pour ces micro-ordinateurs. L'ordinateur était désormais passé du centre de calcul à la maison. C'est alors qu'en raison de l'explosion du marché les fabricants et les utilisateurs commencèrent véritablement à se soucier du problème du logiciel. Encore aujourd'hui, le logiciel (le software) disponible pour micro-ordinateur est très réduit.

1977 devait marquer le début d'un phénomène nouveau. Le marché des Hobbyists était désormais essentiellement saturé. En d'autres termes les utilisateurs n'acceptaient plus les problèmes initiaux des premiers micro-ordinateurs et exigeaient désormais des systèmes complets, fiables et équipés de moyens logiciels minimaux. Dès lors, le potentiel de ces micro-ordinateurs pour les applications professionnelles et commerciales fut révélé... Il était désormais possible d'acheter pour quelques dizaines de milliers de francs un ordinateur dont la puissance de traitement serait désormais suffisante pour assurer la gestion d'une petite entreprise. Ceci est-il véritablement exact ? C'est ce que nous allons établir. Une proportion importante des ventes de micro-ordinateurs aujourd'hui s'adresse désormais à ce marché commercial. Etablissons donc les caractéristiques requises pour une telle application :

Caractéristiques des applications commerciales

Dans une application commerciale, les tâches essentielles de gestion à assurer sont l'automatisation des fichiers clients, fournisseurs, la tenue des comptes courants et des livres de la société. De plus, l'automatisation des ventes, des stocks, pièces détachées et les calculs des salaires ainsi que les diverses déclarations d'impôts sont nécessaires. Ce sont là des tâches traditionnellement confiées aux ordinateurs puissants, c'est-à-dire dans les grandes entreprises. Un exemple fera mieux comprendre le type de l'automatisation que doit fournir un bon système informatique.

Une commande est reçue. Cette commande va être enregistrée par les systèmes et un opérateur va taper les données sur un clavier. La transaction sera alors enregistrée dans un fichier de ventes. Toutefois, ceci n'est pas suffisant. Un système de gestion correctement automatisé va également effectuer toute une série de mise à jour sur d'autres fichiers. Examinons-les. Lorsque la commande est enregistrée, le fichier inventaire va être consulté afin de vérifier la disponibilité de tous les éléments de la

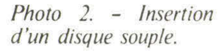


Photo 2. – Insertion d'un disque souple.

commande. Si l'un des objets commandés n'est pas disponible, une commande de réapprovisionnement devra être automatiquement engendrée. Si le stock de pièces disponibles a atteint un niveau fixé à l'avance, une commande de réapprovisionnement sera également délivrée. L'inventaire sera mis à jour en fonction de la commande. Le fichier client sera mis à jour et le montant de la T.V.A. sera stocké dans un fichier séparé de taxe due à l'Etat. De plus, le nom du client sera inséré dans un fichier d'adresses destiné à la prospection commerciale. La facture client sera alors automatiquement imprimée et enfin le montant de la vente sera porté aux journaux comptables appropriés. Il est important de souligner que même pour une transaction de base d'apparence simple, telle qu'une commande reçue, la séquence des mises à jour de fichier peut être tout à fait complexe.

C'est le rôle d'un bon système de gestion automatisée d'effectuer l'ensemble de ces mises à jour dans la séquence appropriée et de manière complète sans la nécessité d'interventions humaines successives.

Examinons à présent de façon plus détaillée les contraintes imposées sur le matériel et sur le logiciel des systèmes à micro-ordinateurs pour satisfaire aux nécessités d'une telle application commerciale ou professionnelle.

Le matériel

Le matériel ou hardware en anglais, se réfère à l'ensemble des composants servant à l'assemblage d'un système. Tout système à micro-ordinateur est composé du micro-ordinateur proprement dit, de terminaux et d'une mémoire de masse. Examinons tour à tour ces composants. Le micro-ordinateur consiste en une boîte équipée d'une alimentation, d'un interrupteur et d'une carte mère dans laquelle toutes les autres cartes sont insérées. La carte essentielle est la carte comprenant le microprocesseur proprement dit qui fournit l'élément de traitement. De plus, le système est équipé de une ou plusieurs cartes memoires. De manière typique pour une application commerciale ou professionnelle, l'ensemble comportera au moins trente deux K ou généralement quarante-huit K de mémoires (moins dans le cas d'une application personnelle). D'autres cartes seront également insérées dans cette boîte permettant de coupler le système à divers périphériques. C'est ainsi que les cartes dites de contrôleurs de disques ou d'écrans de visualisation seront introduites dans le boîtier. La boîte micro-ordinateur peut avoir n'importe quel aspect mais se présente généralement sous la forme d'une boîte rectangulaire. Dans le cas de certains micro-ordinateurs minimaux à usages personnels qui n'utilisent qu'une quantité limitée de mémoires et peu ou pas de cartes interface, le micro-ordinateur proprement dit est à l'intérieur de l'un des autres composants du système tel que par exemple le terminal à écrans de visualisation ou parfois même le clavier.

De manière à pouvoir stocker une quantité suffisante de programmes et des fichiers volumineux, il est indispensable de disposer d'une mémoire de masse. Dans le cas d'applications individuelles, un simple magnétophone à cassettes peut être suffisant. Dans presque tous les cas et toujours dans le cas d'applications commerciales, c'est un disque qui est utilisé. Examinons ici les trois types de disques qui peuvent équiper un système à micro-ordinateur.

Le disque traditionnellement utilisé avec les ordinateurs classiques est le disque dit dur. C'est un disque à larges capacités (plusieurs millions d'octets) d'un temps d'accès relativement rapide, mais très onéreux. Son prix de revient élevé le rend incompatible avec les micro-ordinateurs. Toutefois un type nouveau de mini-disque dur de type Winchester est en voie d'introduction et pourrait devenir rapidement une option dans le cas d'applications commerciales.

Le disque traditionnellement utilisé avec les micro-ordinateurs est le disque souple ou le mini-disque souple (en anglais **floppy disk**). Le disque souple fut inventé à l'origine par Shugart mais est désormais fabriqué par un grand nombre de constructeurs. La disquette proprement dite est souple d'où son nom, elle est recouverte en général d'un seul côté d'un oxyde magnétique sur lequel les informations seront emmagasinées sous forme binaire, c'est-à-dire sous forme d'une succession de 0 et de 1. Cette disquette tourne à grande vitesse à l'intérieur d'un étui en carton. Dans le cas d'un disque souple normal, il s'agit d'une disquette de 8 pouces * et dans le cas d'une mini-disquette le format est de 5,25 pouces. Le mini-disque contient environ trois fois moins de données et a un temps d'accès plus long que les disques souples mais est également disponible à coût beaucoup plus faible. L'intérieur de l'étui en carton est revêtu d'un enduit plastique qui réduit le frottement, lubrifie et nettoie la surface de la disquette. De plus, une ouverture est pratiquée à travers l'une des faces de carton le long d'un des rayons du disque de manière à permettre à la tête de lecture de se déplacer le long de ce rayon. Les données sont stockées sur le disque sous forme de bloc en général 128 MOTS répartis sur les secteurs. De manière à pouvoir accéder d'une façon précise aux données à la surface du disque, tout disque est découpé en pistes et secteurs. Les pistes sont des anneaux concentriques à partir du centre tandis que les secteurs correspondent à des tranches triangulaires du disque. Un disque souple classique a ainsi 77 pistes et 26 secteurs pour le format IBM 3340. De manière à accéder aux données, la tête de lecture se déplace le long d'un des rayons du disque et se positionne au-dessus de la piste voulue puis elle est appliquée à la surface du disque et le mécanisme de lecture ou d'écriture attend le passage du secteur choisi sous la tête de lecture. Un bloc de 128 mots peut alors être transféré. Un disque souple peut ainsi stocker 256 K octets. De plus des versions dites à double densité ou à deux côtés permettent de doubler la capacité d'un tel disque mais souvent aux dépens de sa fiabilité.

Pour des applications indivi-

** 1 pouce = 2,54 cm.*

Pour des applications individuelles, le mini disque souple représente le compromis idéal entre coût et capacité.



Photo 3. – Un système ordinateur inclu le micro-ordinateur, un terminal de type commercial, une imprimante et deux disques.

duelles, le mini-disque souple représente le compromis idéal entre coût et capacité. C'est de loin le type de mémoire de masse le plus utilisé pour le stockage des fichiers et des programmes. Toutefois pour les applications commerciales, il est tout à fait inadéquat en raison de sa capacité insuffisante et de son long temps d'accès. Un disque souple classique est, au minimum, nécessaire. En fait deux disques sont indispensables car il est nécessaire de copier les fichiers ou de trier deux fichiers résidant sur deux disques séparés. En fait même le disque souple classique est encore trop limité et trop lent et c'est véritablement un disque dur qu'il serait bon d'utiliser dans la majorité des applications commerciales. Néanmoins, afin de limiter le coût d'un tel système on peut envisager l'utilisation de disque souple comme un compromis acceptable pour le moment.

Examinons à présent les autres périphériques requis par un système.

L'imprimante

C'est probablement le périphérique le plus important dans le cas d'une application commerciale car le plus onéreux. En effet, pour un usage individuel, une imprimante peut ne pas être nécessaire et de toute manière n'a pas besoin de fournir de performance notable. Son prix de revient est alors tout à fait limité et il peut s'agir d'une télétype traditionnelle. Dans le cas d'applications commerciales, l'imprimante doit être

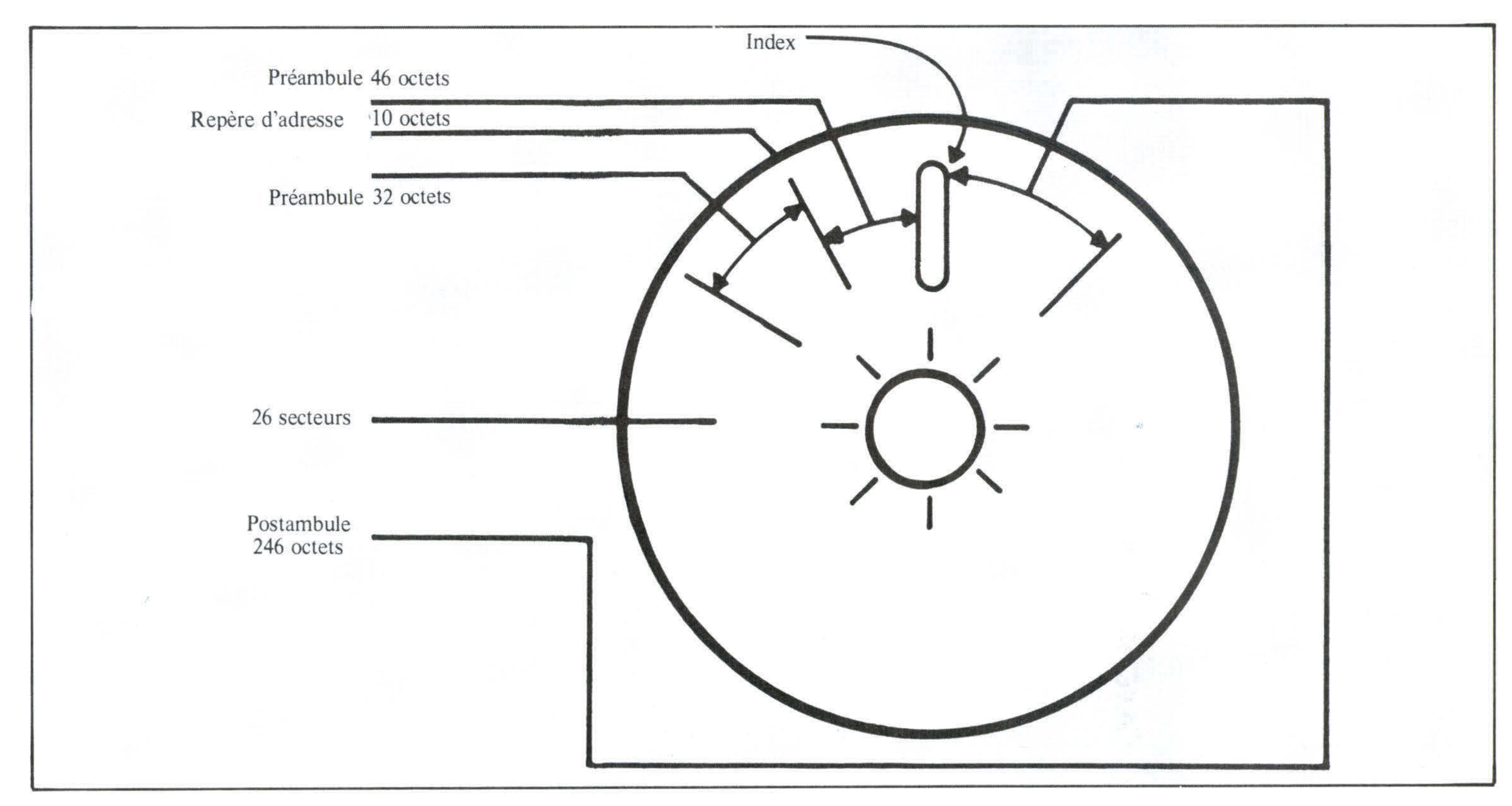


Fig. 1. Décomposition des secteurs dans un disque souple.

capable d'une impression d'au moins 60 caractères par seconde, être d'une grande fiabilité et fournir une qualité d'impression acceptable avec majuscule et minuscule si du courrier commercial doit être imprimé. L'imprimante pour le moment la mieux adaptée à ces demandes est probablement l'imprimante à euromarguerites qui permet à la fois une vitesse acceptable et une haute qualité d'impression. Malheureusement ce type d'imprimante est relativement onéreux, en tous cas plus onéreux que le micro-ordinateur équipé de sa mémoire et des autres périphériques. C'est donc probablement l'un des éléments du système à choisir avec le plus de soins car ce sera l'un des plus durables dans la mesure où l'unité centrale elle-même deviendra rapidement optionnelle. Dans le cas où la qualité d'impression n'est pas essentielle, des imprimantes meilleur marché peuvent être achetées.

Le terminal

Le terminal désormais universel de communication avec un micro-ordinateur est le terminal à écran cathodique. Le consensus est que pour une application commerciale cet écran doit mesurer au moins 12 pouces de diamètre et être capable d'afficher 24 lignes de 80 caractères avec majuscules et minuscules. De plus, pour les applications commerciales le clavier de ce terminal doit être équipé non seulement du clavier alphanumérique classique, tel que celui qui équipe une machine à écrire, mais également d'un clavier accessoire comprenant en particulier l'ensemble des chiffres destinés à la comptabilité, et les touches dites de commande permettant de positionner le curseur sur l'écran ou de modifier le texte durant leur frappe. Toutes les transactions avec le système micro-ordinateur sont normalement effectuées par ce terminal.

Le microprocesseur

Nous n'avons pas mentionné ici comme un critère la nature du microprocesseur utilisé dans la boîte micro-ordinateur, ou sa vitesse d'exécution. En effet, ceci a peu d'importance. Il est exact que la performance du microprocesseur utilisé peut varier dans un rapport de 1 à 5, suivant la nature du composant choisi. Toutefois ceci n'a que peu d'importance dans le cas d'une application commerciale ou professionnelle. En effet, c'est la performance du logiciel qui va conditionner la vitesse d'exécution des programmes d'application.

En effet, pratiquement tous les programmes d'application seront rédigés dans un langage de haut niveau tel que le **Basic.** Pour pouvoir être exécutés ces programmes vont nécessiter la présence d'un interpréteur ou d'un compilateur basic. Hors, la performance de tels interpréteurs ou compilateurs varie dans un rapport de 1 à 100 suivant les systèmes. C'est l'efficacité de ce système software qui va effectivement conditionner la performance d'une application. Dans de nombreux cas d'ailleurs la vitesse d'exécution du programme n'a aucun impact sur la valeur éventuelle du système dans la mesure où l'ordinateur est largement capable d'effectuer toutes les opérations voulues à temps et les limitations sont en général au niveau des entrées et des sorties. Au niveau des entrées elles sont limitées par la vitesse de frappe de l'opérateur, et au niveau des sorties elles sont en général limitées

L'introduction d'un système à micro-ordinateur dans une entreprise est en général marquée par le « choc ordinateur ».

par la relative lenteur de l'imprimante utilisée. La puissance de traitement de l'ordinateur est en général tout à fait suffisante pour traiter les données dans les applications simples de type comptable.

En conclusion, le type de microprocesseur utilisé n'a que peu d'importance pour une application commerciale sauf dans la mesure où il existe, pour certains types de microprocesseurs, plus de logiciel d'utilisation. Toutefois la disponibilité éventuelle d'un bus standardisé tel que le bus S 100 offre des avantages certains tels que la disponibilité de très nombreux autres modules qu'il est facile d'insérer dans le système. Toutefois à nouveau dans le cadre d'une application commerciale, la probabilité que des cartes exotiques soient insérées est très faible et la standardisation peut donc n'avoir que peu de valeur dans la mesure où c'est la disponibilité du software qui est primordiale. Examinons à présent le problème du logiciel.

Le logiciel

Le logiciel est la ressource essentielle qui fait défaut, encore à présent, à tous les systèmes de micro-ordinateurs. Ce n'est que très récemment que les systèmes à micro-ordinateur se sont vus équipés de langage de haut niveau interprété ou compilé par du logiciel performant. Le Basic est devenu le langage prépondérant pour les applications et le système d'exploitation CP/M est devenu un nouveau standard relatif permettant désormais aux utilisateurs d'échanger des programmes ou des fichiers suivant le format utilisé par CP/M. Il est important de souligner ici que toutes les ressources matérielles nécessaires à la réalisation d'un système efficace à vocation personnelle ou commerciale existent. Ce sont les ressources logicielles qui en limitent la performance et l'utilisation. Il n'existe pour le moment aucun système à micro-ordinateur équipé de la totalité des ressources logicielles nécessaires pour l'automatisation complète d'une petite entreprise. Seuls des programmes séparés existent en général tels que des programmes de calcul d'impôts ou des programmes d'enregistrement des ventes, de gestion de stocks ou de mise à jour de fichiers. Toutefois le séquencement automatique nécessaire à la gestion simultanée de tous ces fichiers n'est en général pas réalisé. Il faut donc une série de mises à jour successives par un opérateur humain ce qui limite la valeur éventuelle de tels systèmes. Le logiciel d'applications a des fins commerciales commence à peine à apparaître et l'on peut s'attendre à ce que dans l'année à venir une très grande gamme de logiciels dévienne désormais disponible sur disques et sur cassettes.

Toutefois, même dans l'hypothèse où un grand nombre de programmes distincts à vocation commerciale sont disponibles, sont-ils véritablement applicables à l'entreprise moyenne ?

Ils sont probablement applicables mais ils ne sont pas toujours correctement adaptés. En effet, les systèmes de gestion de chaque petite entreprise varient considérablement d'une entreprise à l'autre. De manière à être généraux dans leur conception, c'est-à-dire à être applicables à un très grand nombre de petites et moyennes entreprises, ces programmes sont encombrants et inefficaces à l'utilisation. Par exemple un programme de gestion de stocks interrogera l'opérateur sur le numéro de pièce, la quantité minimale de réapprovisionnement, le prix d'achat, le prix de vente, les remises éventuelles, les pièces complémentaires que le client devrait commander, les couleurs, et ainsi de suite. En d'autres termes, un très grand nombre d'attributs seront associés à chaque pièce entrée dans le système de gestion de stocks. Il peut en résulter une perte de temps considérable durant la frappe par l'opérateur au terminal et un gaspillage de place sur le disque qui fait en sorte que seul un nombre relativement faible de pièces pourra être stocké sur une disquette. Pour la plupart des applications, il est désirable d'adapter des programmes dits standardisés au cas spécifique de l'organisation qui les utilise. Toutefois, c'est cette adaptation des programmes ou cette optimisation des programmes qui est l'élément le plus coûteux de tout système ordinateur, grand ou petit. Une telle optimisation requiert l'emploi à temps partiel ou à plein temps d'un ou plusieurs programmeurs et coûtera au bout de quelques mois ou d'un an beaucoup plus cher que l'acquisition initiale du système.

De plus, l'introduction d'un système à micro-ordinateur dans une petite entreprise est en général marquée par ce qu'on appelle le « choc ordinateur ». L'introduction de l'ordinateur avec sa discipline de travail nouvelle et le format différent requis pour entrer ou sortir les données résulte en général d'une longue phase de confusions durant laquelle l'entreprise éprouve de nombreuses difficultés. Ce n'est qu'au bout de plusieurs semaines ou plusieurs mois que l'entreprise qui a survécu à ce choc initial sera enfin en mesure d'apprécier les bénéfices présumés de l'ordinateur. Dans la plupart des petites entreprises, l'introduction d'un micro-ordinateur ne supprime aucun poste. La présence du même personnel est toujours nécessaire, tant par sécurité que pour utiliser l'ordinateur. C'est seulement dans l'hypothèse d'une croissance rapide que les bénéfices véritables de l'automatisation par ordinateur se font sentir ; il devient alors possible, à l'aide du même personnel, de faire face à une quantité croissante de transactions. En effet, le traitement automatisé permet de gérer un très grand nombre de pièces détachées ou bien un très grand nombre de transactions avec un personnel identique.

Dans ces conditions, est-il véritablement nécessaire d'envisager l'acquisition d'un système à ordinateur pour une application professionnelle ou commerciale ?

Mythe ou réalité ?

Nous avons désormais établi que l'ensemble des éléments matériels nécessaires à l'informatisation d'un commerce existent. Les moyens logiciels sont encore limités. De plus, les inconvénients de

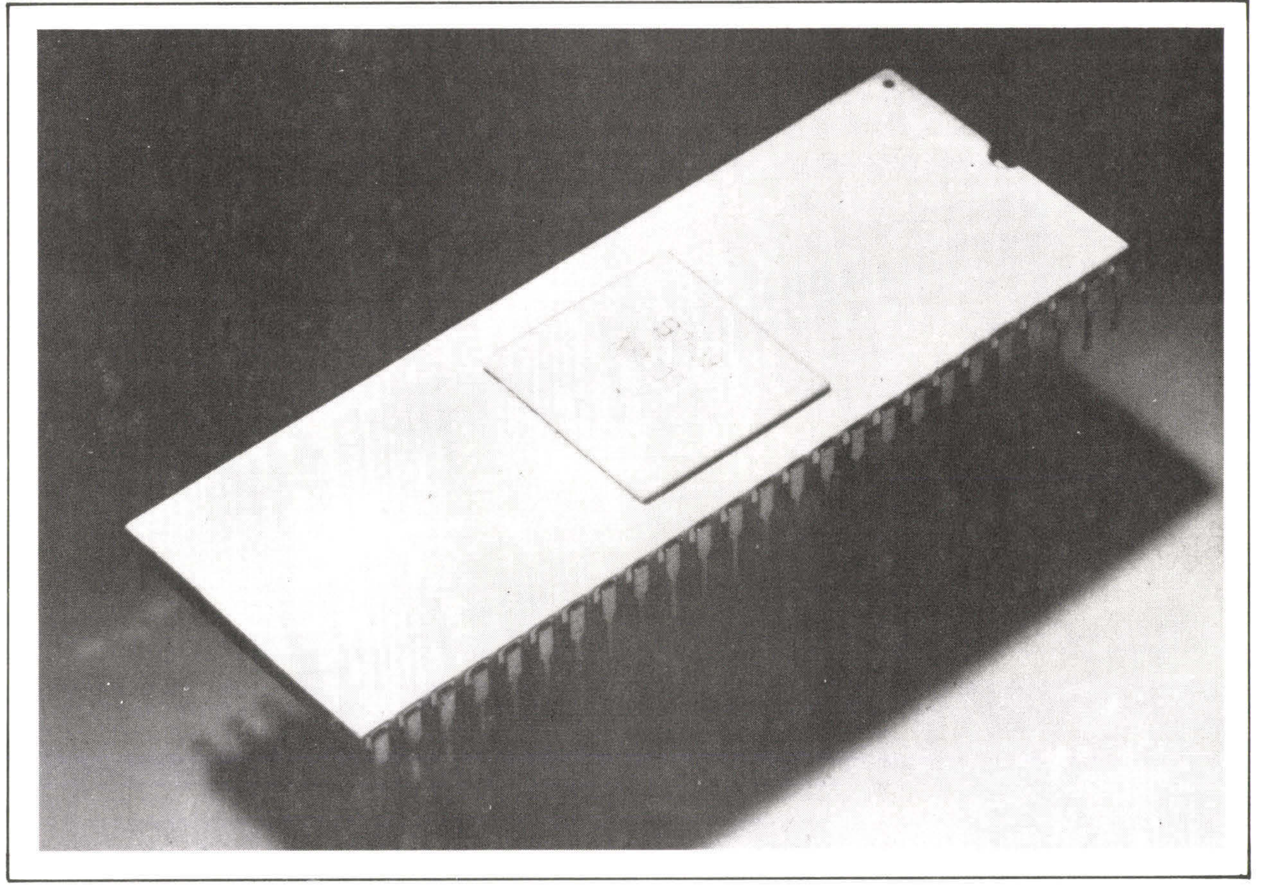


Photo 4. – Le microprocesseur 8X300 de RTC est réalisé en technologie bipolaire schottky. Notez la présence de ses 50 broches. (Doc. RTC.)

cette informatisation sont apparents et vont en général créer une perte initiale plutôt qu'un profit. Toutefois dans toute entreprise en voie d'expansion, l'automatisation d'un nombre même simple des tâches habituelles de gestion va en général apporter un bénéfice largement supérieur au coût d'un tel système. En effet, le coût d'un système est désormais très faible de l'ordre de quelques dizaines de milliers de francs et tant que des programmes complexes de gestion sur mesure ne seront pas nécessaires, il est possible de bénéficier rapidement d'un fichier d'adresses informatisé, d'un fichier clients automatisé avec vérification des bons et des mauvais payeurs, et ainsi de suite. Il ne faut pas espérer que le système va automatiser l'ensemble de la gestion de l'entreprise. Il en est matériellement capable mais ne dispose pas encore des ressources logicielles nécessaires.

D'autre part un avantage essentiel d'une telle informatisation de l'entreprise est sa valeur éducative pour les responsables de la société. En effet, il est désormais possible d'appliquer à l'ordinateur un adage bien connu dans le cas des voitures. Dans toute machine particulièrement complexe, il importe d'apprendre à la maîtriser de la même manière qu'il importe d'apprendre à conduire une automobile. Il est évidemment désirable, tant que l'on ne sait pas bien conduire d'utiliser un véhicule d'occasion ou du moins très bon marché. Ce principe est exactement applicable à l'utilisation d'un ordinateur. L'acquisition d'un micro-ordinateur aujourd'hui malgré son potentiel logiciel limité permet désormais aux responsables de s'initier rapidement à l'utilisation de l'outil informatique et d'apprendre à le maîtriser étape par étape. De plus, la transition de l'entreprise sur un système informatisé peut également être effectuée pas à pas avant que le volume d'affaires ne devienne prohibitif et ne crée ce fameux choc ordinateur. L'utilisateur d'un tel système doit accepter le fait que l'unité centrale qu'il achète sera vraisemblablement rapidement obsolète. Toutefois il sera désormais en mesure une ou plusieurs années plus tard de spécifier précisément par lui-même l'ensemble des données techniques du système qu'il envisage d'acheter en tant que successeur de ce premier système. Il aura littéralement appris à conduire l'ordinateur.

En conclusion, un système individuel à vocation commerciale ou professionnelle ne peut être présenté aujourd'hui comme un système de gestion complet. Le logiciel nécessaire n'est pas encore disponible. Toutefois, il permet de bénéficier tout de suite des avantages d'une informatisation limitée et de s'initier pour la première fois à très faible coût à la gestion par ordinateur. Un tel système par les bénéfices qu'il offre rapidement compense très vite son prix d'achat. Ce serait une erreur grave de ne pas envisager son acquisition dans toute entreprise ou activité professionnelle en expansion.

Rodnay Zaks*

Rodnay Zaks est président de la société Sybex et auteur de nombreux ouvrages sur les microprocesseurs et micro-ordinateurs à usage personnel.



Informatic Systèmes TéléCom

DISTRIBUTEURS RECHERCHÉS DANS TOUTE LA FRANCE

MICRO-ORDINATEUR COMPUCOLOR

- MICROPROCESSEUR 8080 A.
- MEMOIRE RAM DE 8 A 32 Ko.
- BASIC 16 K RESIDENT
- INTERFACE RS 232 C.
- ECRAN DE VISUALISATION (64 c x 16/32).
- 8 COULEURS.
- TRACE GRAPHIQUE.
- 1 UNITE DE MINI-DISQUETTE.
- 64 CARACTERES SPECIAUX.
- OPTION IMPRIMANTE.

PRIX DE VENTE : 11.800 F HT.
COMPRENANT : ECRAN DE VISUALISATION AVEC 8 COULEURS – 1 UNITE DE MINI-DISQUETTE INTEGREE – CLAVIER COMPLET – 8 K RAM – BASIC RESIDENT.



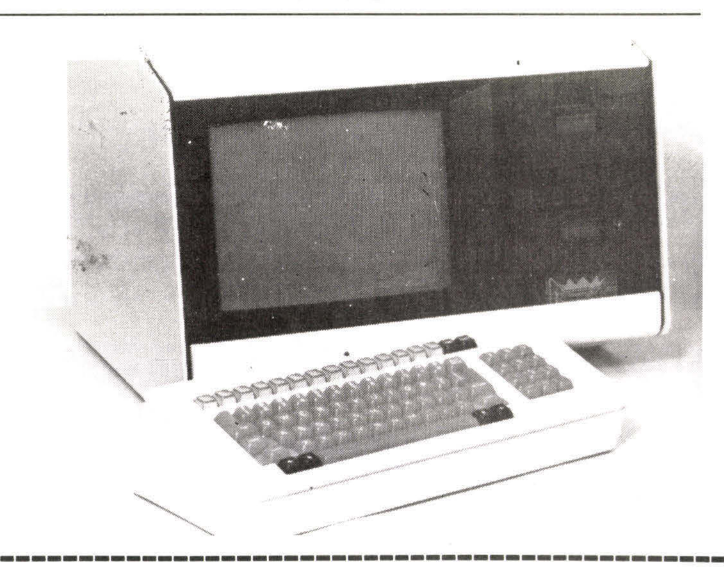
MICRO-ORDINATEUR APPLE-II

- MICROPROCESSEUR ROCKWELL 6502 - RAM EXTENSIBLE DE 4 K A 48 K.
- BASIC - MONITEUR - ASSEMBLEUR - DESASSEMBLEUR (ROM).
- SORTIE VIDEO 24 LIGNES/40 COLONNES.
- GRAPHIQUES FINS EN COULEURS SUR T.V. (RVB-SECAM).
- INTERFACES MAGNETOPHONE ET ENTREES ANALOGIQUES - HAUT-PARLEUR INCORPORE.
- 8 PERIPHERIQUES CONNECTABLES DONT :
  - IMPRIMANTE, MODEM, CARTE DE COMMUNICATION RS 232
  - CARTE DE RECONNAISSANCE VOCALE (32 MOTS QUELCONQUES)
  - FLOPPY DISQUES (1 A 14 FOIS 116 Ko).

* DOS : FICHIERS DE DONNÉES EN ACCÈS SÉQUENTIEL INDEXÉ PROGRAMMATHIQUE/CHAINAGE DES PROGRAMMES/PROTECTIONS D'ÉCRITURE.

MICRO-ORDINATEUR I.S.T.C. 5000

- MICROPROCESSEUR Z80 - RAM de 32 K à 64 K.
- SORTIE VIDEO 24 LIGNES/80 COLONNES.
- GENERATEUR DE CARACTERES PROGRAMMABLE.
- 1 OU 2 MINI-FLOPPY DISQUES (DOUBLE FACE) INTEGRES.
- DOS-EDITEUR DE TEXTE.
- MACRO ASSEMBLEUR.
- BASIC ETENDU (IF THEN ELSE, WHILE, PRINTUSING).
- FORTRAN IV ANSI
- EDITEUR DE LIENS POUR MODULES FORTRAN.
- 2 A 5 CONNECTEURS BUS S-100.
- INTERRUPTIONS CHAINEES AVEC PRIORITES (8 NIVEAUX).
- CARTE DE COMMUNICATION (SYNCHRONE/ASYNCHRONE).
- IMPRIMANTE AVEC INTERFACE.



Micro-Systèmes

Bon réponse à retourner à : I.S.T.C., 7 à 11, rue Paul-Barruel, 75015 Paris. Tél. : 306.46.06.

Raison Sociale ....

Activité ....

Nom et Fonction ....

....

Intéressé par : COMPUCOLOR ☐
APPLE II ☐ I.S.T.C. 5000 ☐

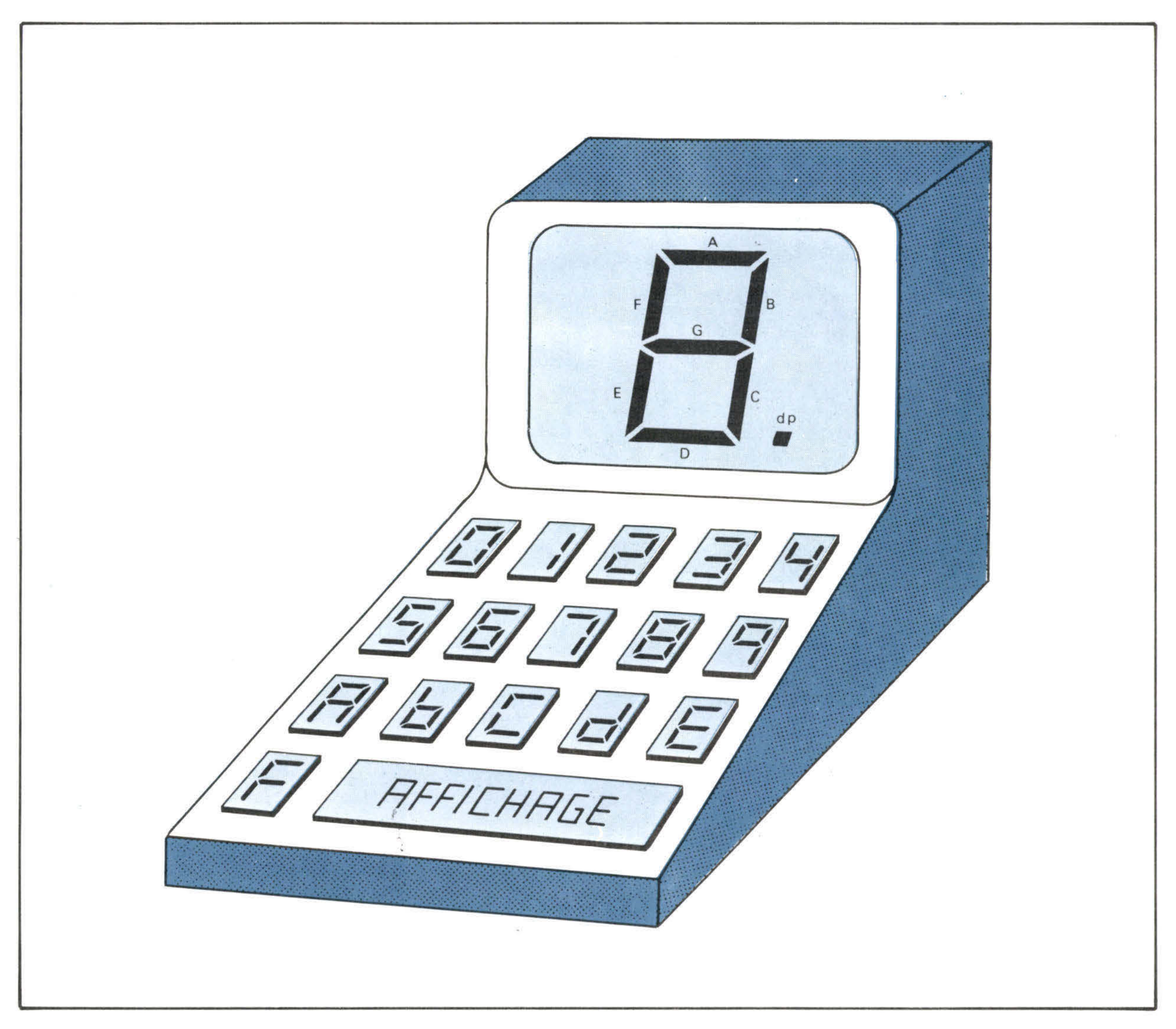
Adresse ....

....

Tél ....

☐ UNE DEMONSTRATION ☐ LA VISITE D'UN COMMERCIAL

L'affichage hexadécimal



* Fairchild : Division S.C., 121, avenue d'Italie, 75013 Paris. Tél. : (16-1) 584.55.66.

Depuis l'avènement des microprocesseurs, l'affichage hexadécimal a pris une importance toute particulière. De fait, il permet de générer les symboles correspondant aux seize combinaisons possibles d'une information sur 4 bits puisque cette base de numération est 16 (2^4), au lieu de 10 pour le décimal et 8 (2^3) pour l'octal.

Récemment, alors que je travaillais sur un périphérique de mon système à microprocesseur, une intéressante question me vint à l'esprit : « Quelle est la façon la moins chère de visualiser en hexadécimal une donnée sur un panneau d'affichage ? » Ce ne devrait pas être trop difficile de répondre à cette question, regardons ce que nous pouvons trouver dans les catalogues des différents fabricants.

Cet affichage sera très probablement du type 7 segments à diode électroluminescente ou LED (Light Emitting Diode), car il existe de très nombreux types d'afficheurs tous facilement disponibles.

Le boîtier que j'utilise le plus fréquemment est le 5082-7730/31 de Hewlett-Packard. Commercialisé en boîtier du type MAN-1, sa luminosité est très bonne et il peut se lire aisément à une distance de 3 m ! C'est un afficheur à anode commune ; la **figure 1 a** représente son circuit de commande. La même configuration pour des LED à cathode commune est donnée **figure 1 b.**

Quels sont les décodeurs existants ?

Avant toutes choses, regardons dans la famille logique 7400. Il en existe de très nombreux d'un coût relativement faible.

Parmi la liste des décodeurs-drivers BCD/7 segments nous notons les circuits 7446, 7447, 7448 et 7449 bien connus. Ces dispositifs décodent une entrée BCD et génèrent la tension de sortie nécessaire à un afficheur 7 segments **(fig. 2 a).** Remarquez que les nombres 0 à 9 correspondent bien au code souhaité, mais que les lettres A à F ne sont pas représentées (à la place de celles-ci nous avons des symboles quelconques). Que pouvons-nous trouver de mieux adapté à nos besoins ?

Le catalogue de Fairchild * indique des décodeurs intéressants et de plus deux décodeurs hexadécimaux désignés par la référence 9368 et 9370. Ce sont des latch/décoder/driver qui affichent les symboles 0... 8, 9, A, B... F de la **figure 2 b.** Le 9368 peut commander directement un afficheur

Une mémoire programmable (PROM) de 32 x 8 bits peut constituer un convertisseur BCD/7 segments.

+5 VDC
1 KΩ
1/4 W
Affichage inhibé
5V "1"
0V "0"
Affichage autorisé
1 KΩ
1/4 W
2N2907
ou
2N3906
I Max.
Afficheur LED
à anode
commune
Lignes "segments"
220Ω
1/4 W
A B C D E F G DP
Décodeur
7 segments
pour anodes
communes
Vers les segments
des autres afficheurs
dans une configuration
multiplexée
Segment éteint
Valeur pour courant
de 15 mA/segment
"1"
"0"
Segment allumé

Fig. 1a. – Configuration du circuit de commande d'un afficheur LED à anode commune.

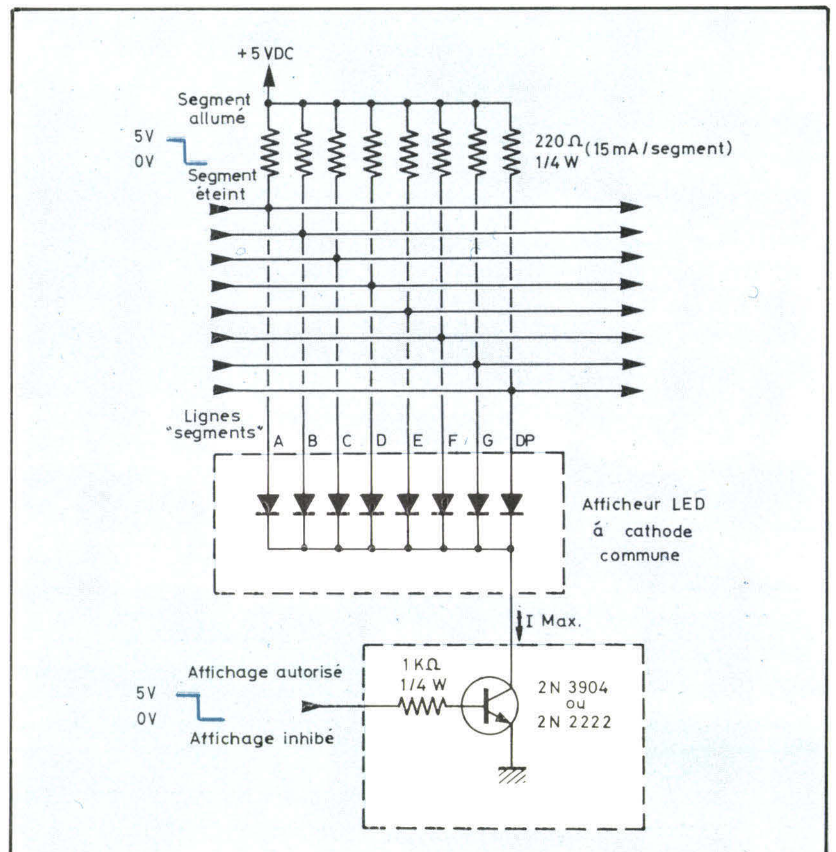


Fig. 1b. – Circuit de commande d'un afficheur à cathode commune.

LED à cathode commune puisqu'il délivre un courant maximum de 20 mA/segment. Le 9370, en tout point identique au 9368, est destiné aux afficheurs à anode commune. Le brochage de ces deux circuits est représenté **figure 3.**

L'afficheur Dialight 745-0007 est le circuit suivant que nous découvrons dans les catalogues. Il intègre dans un même boîtier DIL 14 broches une mémoire (latch), un décodeur-driver et un affichage LED hexadécimal dont les symboles sont donnés **figure 2c.** La **figure 4** indique le brochage de ce circuit. Néanmoins, bien que très intéressant, ces dispositifs sont d'un prix assez élevé et le coût d'un panneau de 8 chiffres devient rapidement prohibitif. Cherchons une autre solution.

Réalisation d'un décodeur

Pourquoi ne pas réaliser nous-même un décodeur/driver ? Nous pouvons utiliser une PROM pour convertir le code binaire BCD de

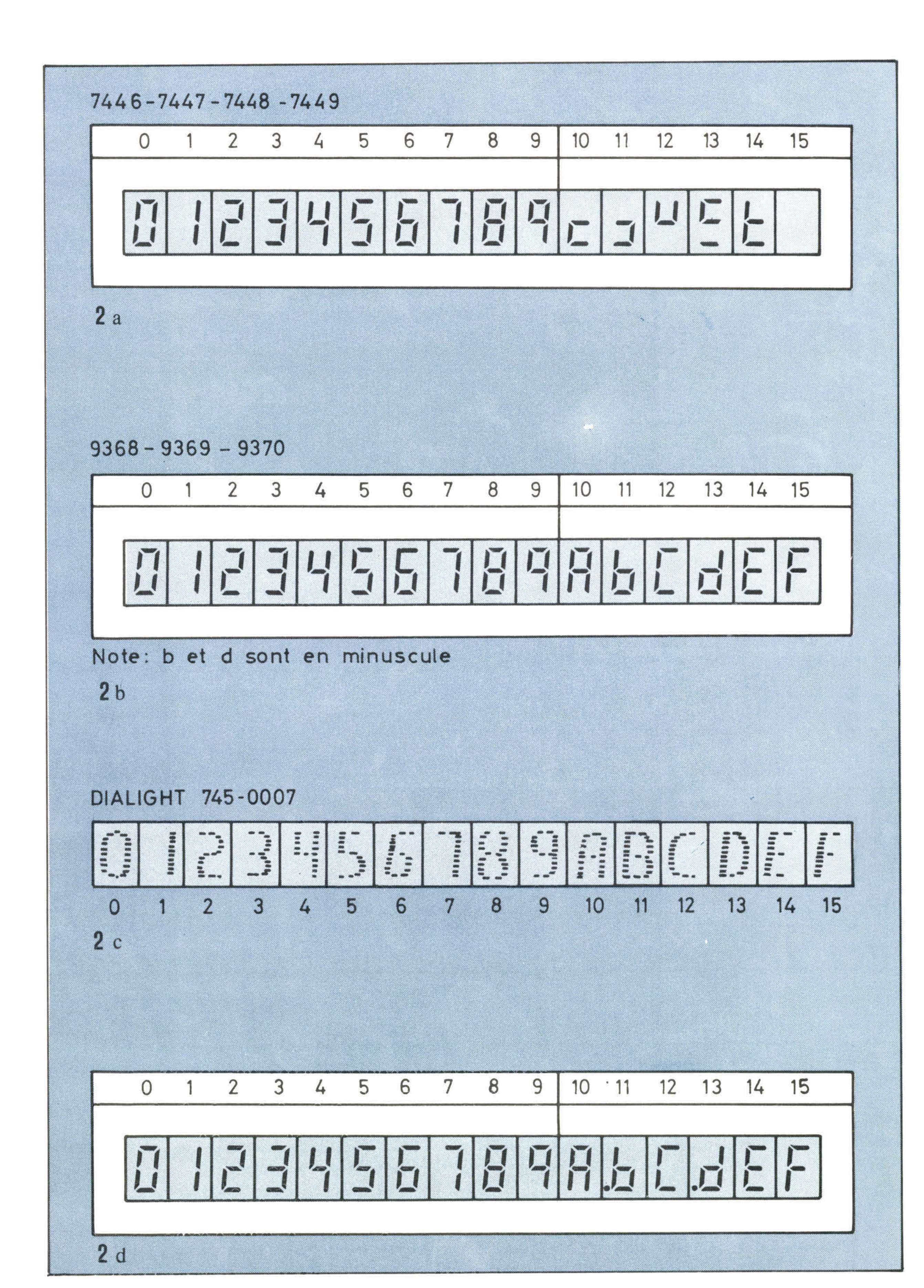


Fig. 2. – Différents formats d'affichage issus de décodeurs classiques.

Chaque fois que le nombre de boîtier afficheur dépassera 2 ou 3, il faudra réaliser un affichage multiplexé

** « Extinction » : Blanking Input ou BI en anglo-saxon.*

** $\overline{CE}$: validation de la mémoire. Entrée active au niveau 0.*

4 bits d'entrée en une information de sortie compatible avec un afficheur 7 segments classique.

Prenons, par exemple, la mémoire 82S23. C'est une mémoire programmable (PROM) de 256 bits organisée en 32 mots de 8 bits. Ses sorties sont à collecteur ouvert, un bon choix pour la commande de LED **(fig. 5).**

Bien entendu, une PROM 32 x 8 aura en entrée 5 lignes d'adresses ($2^5 = 32$) et 8 lignes de données en sortie. Quatre de ces cinq lignes d'adresses (A_0, A_1, A_2, A_3) seront les 4 bits d'entrée binaire A B C D et sept des huit sorties de données devront jouer le rôle de commande des 7 segments.

Pouvons-nous employer les lignes restantes ? Réfléchissons. La ligne d'adresse A_4 sélecte les 16 premiers ou derniers mots mémoires de 8 bits. Lorsque nous programmons un décodeur hexadécimal dans la deuxième moitié de la PROM (A_4 au niveau logique « 1 ») et uniquement ces « 0 » dans la première moitié, puis si nous faisons passer A_4 de « 1 » logique au « 0 » logique, tous les segments seront allumés quel que soit le code présent sur les entrées A_0-A_3. Lorsque nous utilisons un afficheur à anode commune. La ligne A_4 devient alors notre entrée « lampe test ».

Dans le cas où un afficheur à cathode commune intervient dans une réalisation, cette ligne sera la ligne d'extinction de tous les segments. La ligne A_4 sera ici l'entrée « extinction* ».

Voyons maintenant le rôle de l'entrée « sélection de la mémoire » ($\overline{CE}$: Chip Enable). Avec $\overline{CE}$ * au niveau logique « 0 », les données adressées par A_0-A_4 sont présentes à la sortie sur B_0-B_7. Si $\overline{CE}$ passe au niveau logique « 1 », les sorties sont déconnectées (tous les collecteurs sont ouverts), $\overline{CE}$ devient alors notre entrée « extinction » ou (BI) lorsque la PROM est utilisée avec un dispositif à anode commune. Quand l'afficheur est du type cathode commune, $\overline{CE}$ est notre « lampe test ».

Le tableau suivant récapitule les actions de $\overline{CE}$ et A_4 pour les types d'afficheurs à anode ou à cathode commune.

	Anode commune	**Cathode commune**
A_4	« Lampe test »	« Extinction » ou B.I.
$\overline{CE}$	« Extinction » ou BI	« Lampe test »

Fig. 3. – Brochage des circuits intégrés « Fairchild » 9368 et 9370. Le 9368 est un latch décodeur/driver hexadécimal pour afficheurs à cathode commune. Le 9370 est destiné aux afficheurs à anode commune.

		9368 / 9370		
A_1	1		16	Vcc
A_2	2		15	F
$\overline{EL}$	3		14	G
$\overline{RB0}$	4		13	A
$\overline{RBI}$	5		12	B
A_3	6		11	C
A_0	7		10	D
0V	8		9	E

Maintenant, il ne nous reste plus qu'à programmer correctement la PROM en fonction des signes choisis.

Les symboles alphanumériques de la **figure 2 b** sont une bonne solution mais le b peut être considéré comme un 6 sans barre supérieure. Pour cette raison, mon choix sera celui de la **figure 2 d.** Le point décimal, commandé par la ligne B_0 de la PROM lève toute ambiguïté en ce qui concerne les lettres b et d.

Ainsi, le listing de programmation du **tableau 1** est étudié en fonction du format spécifique des symboles de la **figure 2 d.** La référence de chacun des 7 segments est indiquée sur l'illustration du titre.

Le **tableau 1** permet de programmer une PROM de commande d'un affichage LED à anode commune.

Le **tableau 2** est destiné plus particulièrement aux afficheurs à cathode commune.

Notez que dans ces deux tableaux toutes les adresses sont en hexadécimale et les données en hexadécimale et en binaire. Il existe de nombreuses mémoires PROM de 256 bits compatibles broches à broches et disponibles.

Nous pouvons citer quelques unes d'entre elles : les 8223 et 82S23 de Signetics, la 8256 de Harris, la 6330-1 de MMI... Mais elles sont souvent compatibles uniquement dans le mode lecture. Chacune de ces mémoires peut être

Tableau 1

Adresse (en Hexa)	Données (en Hexa)	DONNÉE EN BINAIRE								Symboles
		B7	B6	B5	B4	B3	B2	B1	B0	
		Segment								
		a	b	c	d	e	f	g	DP	
00 à 0F	**00**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
10	03	0	0	0	0	0	0	1	1	0
11	9F	1	0	0	1	1	1	1	1	1
12	25	0	0	1	0	0	1	0	1	2
13	0D	0	0	0	0	1	1	0	1	3
14	99	1	0	0	1	1	0	0	1	4
15	49	0	1	0	0	1	0	0	1	5
16	41	0	1	0	0	0	0	0	1	6
17	1F	0	0	0	1	1	1	1	1	7
18	01	0	0	0	0	0	0	0	1	8
19	19	0	0	0	1	1	0	0	1	9
1A	11	0	0	0	1	0	0	0	1	A
1B	C0	1	1	0	0	0	0	0	0	B
1C	63	0	1	1	0	0	0	1	1	C
1D	84	1	0	0	0	0	1	0	0	D
1E	61	0	1	1	0	0	0	0	1	E
1F	71	0	1	1	1	0	0	0	1	F

Tableau 2

Adresse (en Hexa)	Données (en Hexa)	DONNÉE EN BINAIRE								Symboles
		B7	B6	B5	B4	B3	B2	B1	B0	
		Segment								
		a	b	c	d	e	f	g	DP	
00 à 0F	00	0	0	0	0	0	0	0	0	
10	FC	1	1	1	1	1	1	0	0	0
11	60	0	1	1	0	0	0	0	0	1
12	DA	1	1	0	1	1	0	1	0	2
13	F2	1	1	1	1	0	0	1	0	3
14	66	0	1	1	0	0	1	1	0	4
15	B6	1	0	1	1	0	1	1	0	5
16	BE	1	0	1	1	1	1	1	0	6
17	E0	1	1	1	0	0	0	0	0	7
18	FE	1	1	1	1	1	1	1	0	8
19	E6	1	1	1	0	0	1	1	0	9
1A	EE	1	1	1	0	1	1	1	0	A
1B	3F	0	0	1	1	1	1	1	1	B
1C	9C	1	0	0	1	1	1	0	0	C
1D	7B	0	1	1	1	1	0	1	1	D
1E	9E	1	0	0	1	1	1	1	0	E
1F	8E	1	0	0	0	1	1	1	0	F

programmée par différentes méthodes judicieusement choisies. La **figure 5** indique le brochage typique d'un décodeur/driver 7 segments à PROM.

Dans le cas souvent réalisé en pratique où nous avons plusieurs boîtiers d'affichage à commander, deux solutions peuvent être envisagées : placer autant de décodeurs que d'afficheurs ou multiplexer les boîtiers.

L'affichage multiplexé

L'emploi d'un décodeur associé à chaque afficheur est économique si les boîtiers n'excèdent pas 2 ou 3. Lorsque le nombre de symboles à visualiser est supérieur, il sera plus rentable de multiplexer l'affichage.

Notez que le multiplexage n'économise pas uniquement les composants, il réduit aussi notablement la consommation de puissance nécessaire à l'alimentation des LED.

Dans une configuration multiplexée, les segments de tous les afficheurs sont reliés respectivement entre eux de façon à constituer un « BUS segment » **(fig. 6)**. Dès que les informations sont présentes sur ce BUS, le boîtier corres-

Fig. 4. – Brochage de l'afficheur/latch/décodeur/ driver hexadécimal Dialignt 745-0007.

Fig. 5. – Brochage de la PROM : 8223, 82S23 de Signetics, 6330-1 MMI ou 8256 Harris. La signification des entrées A4 et $\overline{CE}$ en fonction du type d'afficheur à commander est indiquée dans le texte.

Fig. 6. – Dans un affichage multiplexé, les segments de tous les afficheurs sont reliés respectivement entre eux.

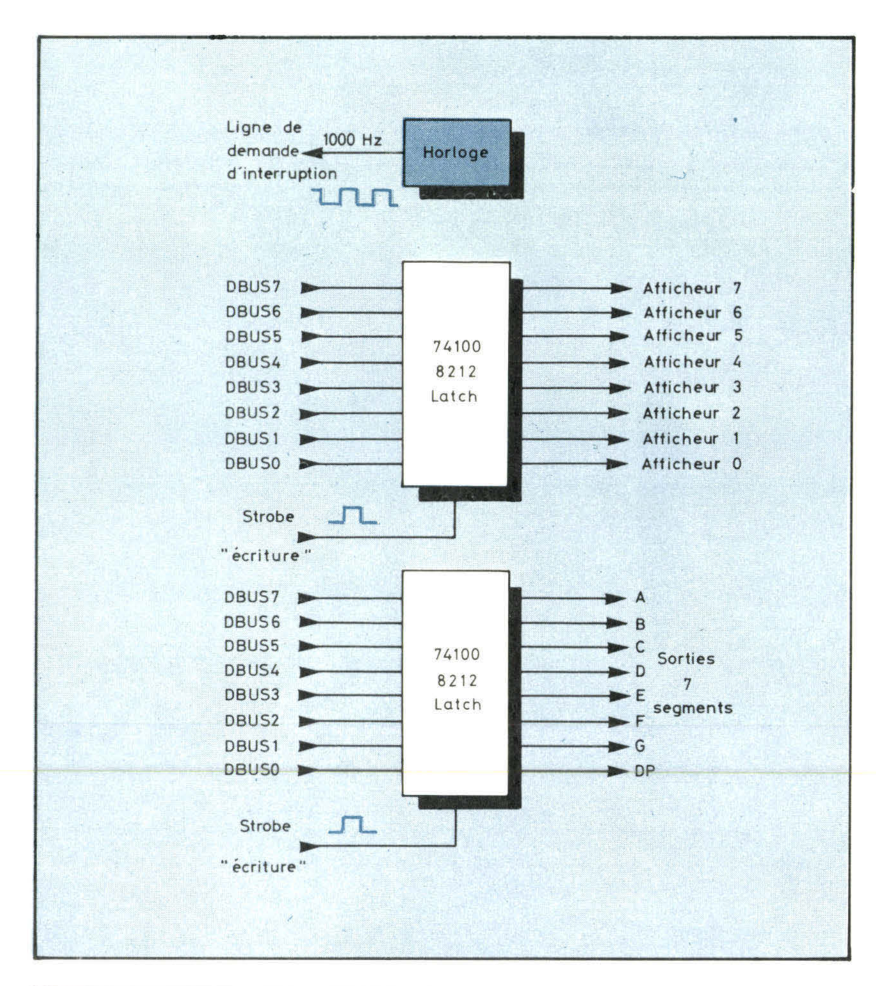


Fig. 7. – Configuration minimale d'un affichage multiplexé. Le microprocesseur traite la conversion binaire BCD/7 segments.

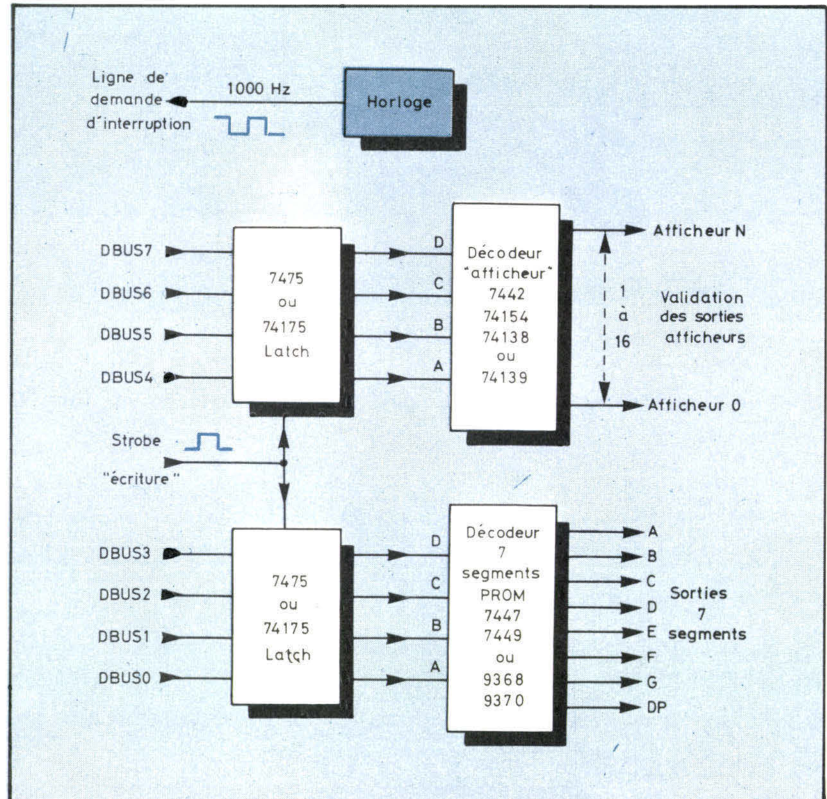


Fig. 8. – Un décodeur pour les afficheurs et un autre pour les segments réduit le programme de traitement. (Le microprocesseur réalise toujours le multiplexage).

pondant est sélectionné. Grâce à une validation séquentielle de chaque afficheur associée aux données qui lui sont propres, les symboles apparaissent en continu.

Ce dispositif nécessite néanmoins un « rafraîchissement » périodique de l'information au rythme de 1000 périodes/s, c'est-à-dire 1 kHz, ce qui évite les effets de scintillement.

Une des méthodes les plus simples de réaliser un multiplexage d'affichage pour microprocesseur, serait de laisser celui-ci gérer à la fois le multiplexage et la conversion binaire/7 segments uniquement par programme **(fig. 7).** La conversion par programme réduira le nombre de composants mais augmentera la taille du logiciel de traitement et son temps d'exécution.

Notez que l'horloge est ici reliée directement à la ligne d'interruption qui occasionnera un saut dans un sous-programme de service « affichage ».

Ce sous-programme devra charger simultanément, et ceci toutes les millisecondes, l'adresse de l'afficheur et la donnée correspondante.

L'étape suivante de notre investigation déplace le traitement de la conversion binaire/7 segments du logiciel au matériel (du soft au hard). C'est ici que notre PROM intervient **(fig. 8).**

La même séquence d'interruption sera utilisée mais l'actuel mot binaire de 4 bits représentant les données devra être mémorisé (latch), au lieu de le convertir directement en informations 7 segments comme dans l'exemple précédent.

Un autre type de multiplexeur peut être envisagé si l'on désire encore réduire la taille du logiciel et simplifier le programme de gestion **(fig. 9).** Ici le microprocesseur n'exécute pas le rafraîchissement. Tous les microprocesseurs doivent écrire les données de 4 bits dans la mémoire de donnée appropriée et c'est alors le hardware qui traite la fonction de multiplexage.

Le multiplexeur de la **figure 10** est encore meilleur car il utilise moins de hardware.

La mémoire RAM 7489 ou

Fig. 9. – Le multiplexage est ici généré par le Hardware. Le microprocesseur n'a plus à « rafraîchir » l'affichage.

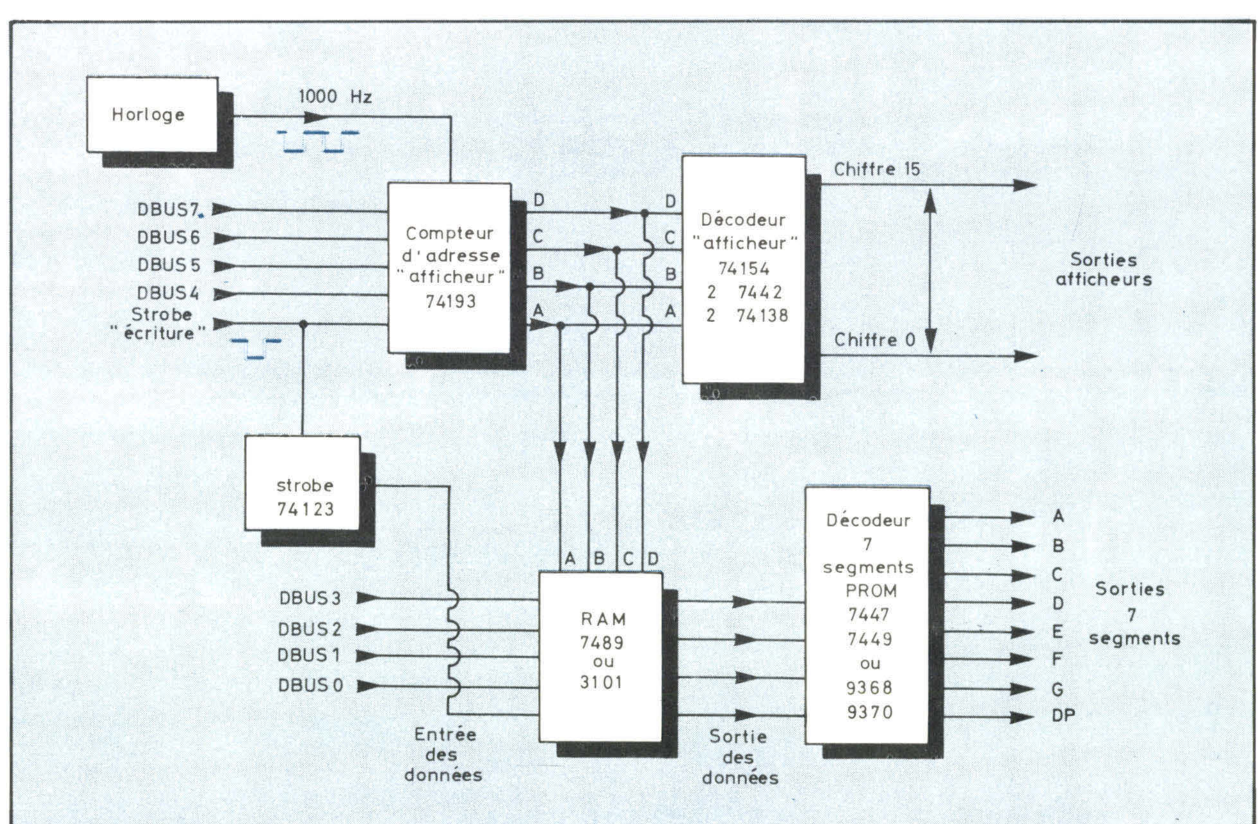
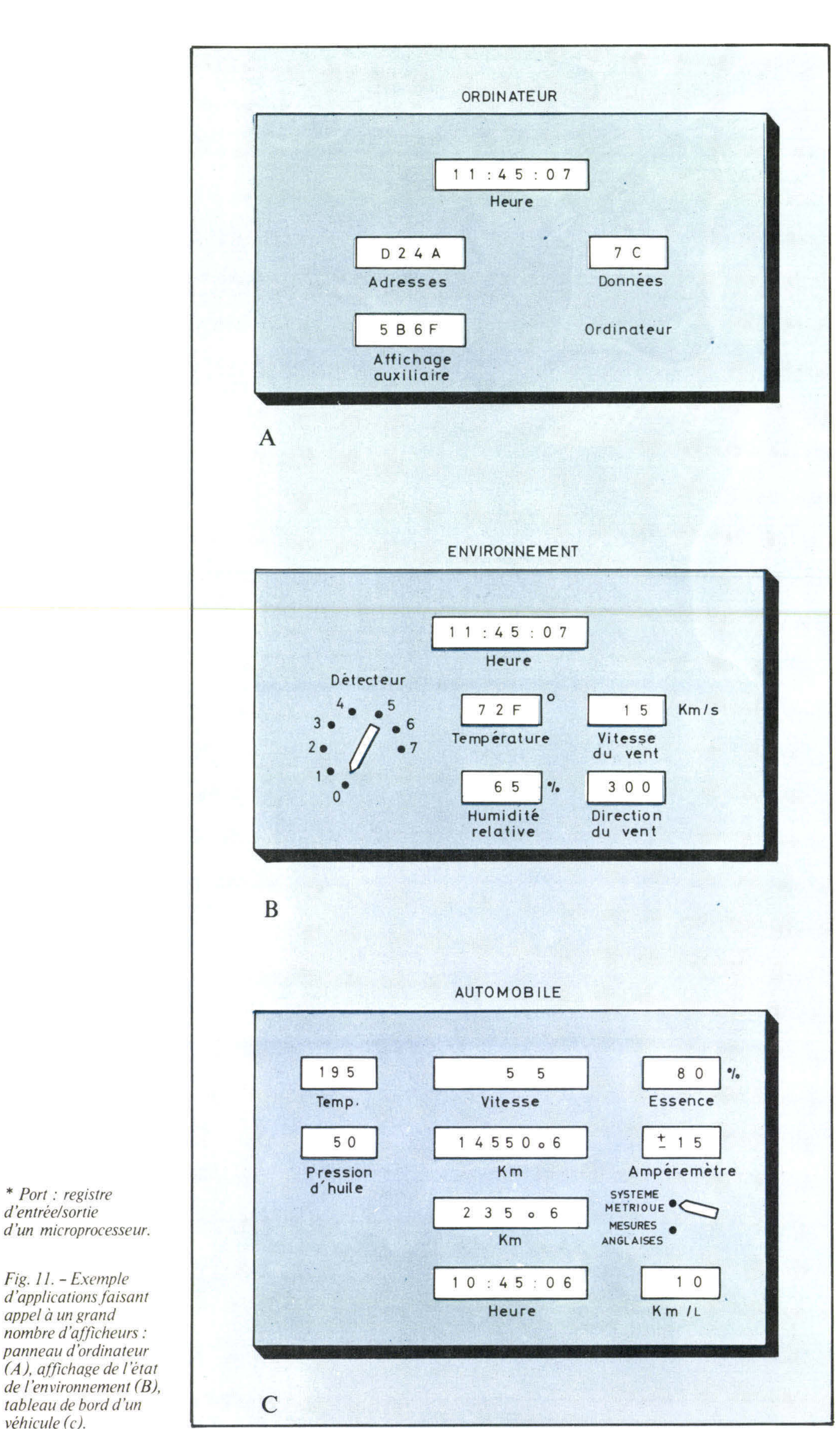


Fig. 10. – Le nombre d'afficheurs augmente et la taille du hardware diminue. Ceci est un exemple d'un affichage multiplexé optimum pour microprocesseur.



* *Port : registre d'entrée/sortie d'un microprocesseur.*

Fig. 11. – Exemple d'applications faisant appel à un grand nombre d'afficheurs : panneau d'ordinateur (A), affichage de l'état de l'environnement (B), tableau de bord d'un véhicule (c).

3101 organisée en 16 mots de 4 bits stockera et réalisera le multiplexage d'informations de 16 afficheurs et ceci dans un boîtier 16 broches.

Ces circuits sont d'ailleurs spécialement conçus pour être commandés d'un port * de 8 bits parallèles.

Applications

Maintenant que nous savons comment réaliser un système d'affichage économique, que pouvons-nous en faire ?

Regardons la **figure 11 a,** elle représente la face avant d'un micro-ordinateur. Outre l'affichage habituel de l'adresse courante et de la donnée vous avez la possibilité de placer un affichage auxiliaire de 16 bits (4 digits) et une visualisation de l'heure sur 6 digits en utilisant un affichage en 16 boîtiers multiplexés.

De même, le panneau de la **figure 11 b** montre un exemple typique de la mesure des conditions atmosphériques de l'environnement. Vous pouvez lire aisément la température (celle-ci pouvant être issue de plusieurs capteurs) en degré Celsius, Fahrenheit ou Kelvin, le pourcentage d'humidité relative, la vitesse et la direction du vent et, bien sûr, l'heure ; le tout sous le contrôle d'un processeur qui gère un multiplexeur 16 digits et une mémoire. Cette idée pourrait être aisément développée lors de l'étude de votre système de chauffage solaire !

Le dernier exemple **(fig. 11 c)** est plus futuriste : le tableau de bord digital d'une automobile. Les données affichées pourront indiquer la vitesse, la distance totale effectuée par le véhicule, la distance journalière, le niveau d'essence (en pourcentage de la capacité), la température (du moteur, de la transmission...), la pression d'huile, le courant délivré par l'alternateur, la consommation en temps réel (en km/l) et, bien entendu, l'heure. Un commutateur peut en outre permettre le choix des unités : système métrique ou mesures anglo-saxonnes. ■

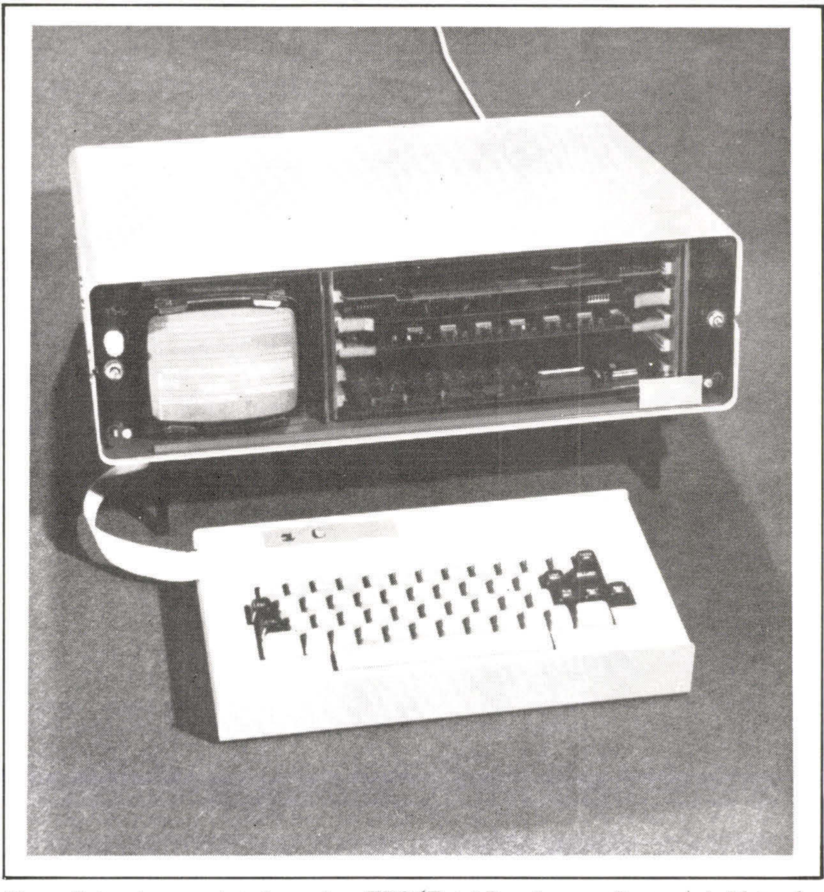
Traduit de Kilobaud, avril 78.

Initiation aux MICROPROCESSEURS

Nous décrivons successivement, dans cet article, le fonctionnement d'un système à microprocesseur en analysant le cheminement et les différentes étapes du traitement des informations. L'adresse d'une instruction étant appliquée au bus d'adressage, l'instruction correspondante est reçue sur le bus de données au cours du cycle d'acquisition, puis décodée. Les transferts d'informations sont effectués conformément à l'instruction décodée et le compteur ordinal fournit l'adresse de l'instruction suivante au microprocesseur.

Nous introduirons, ensuite, les notions d'organigramme, de programme et de langage de programmation pour définir ce que sont compilateur, interpréteur, macro-assembleur, cross-assembleur, assembleur résident, cross-compilateur et même cross-macro-assembleur.

Finalement, nous aborderons les différentes phases de développement d'une application à base de microprocesseurs, la nécessité d'utiliser un micro-ordinateur à structure modulaire et logiciel résident ainsi que des considérations d'ordre financier : coût d'un programme terminé et opérationnel, investissements nécessaires au système matériel et à son logiciel d'accompagnement.



Un outil de mise au point : le système TDS (Total Development System) de Motorola.

Fig. 1. – Représentation symbolique d'un micro-ordinateur avec microprocesseur, mémoires vive et morte et circuit d'interface. La position des aiguillages permet de définir le cheminement des informations lors de l'exécution des différentes séquences d'un programme.

Le compteur ordinal est tenu à jour par la logique de commande et dirige le microprocesseur à chaque étape du programme.

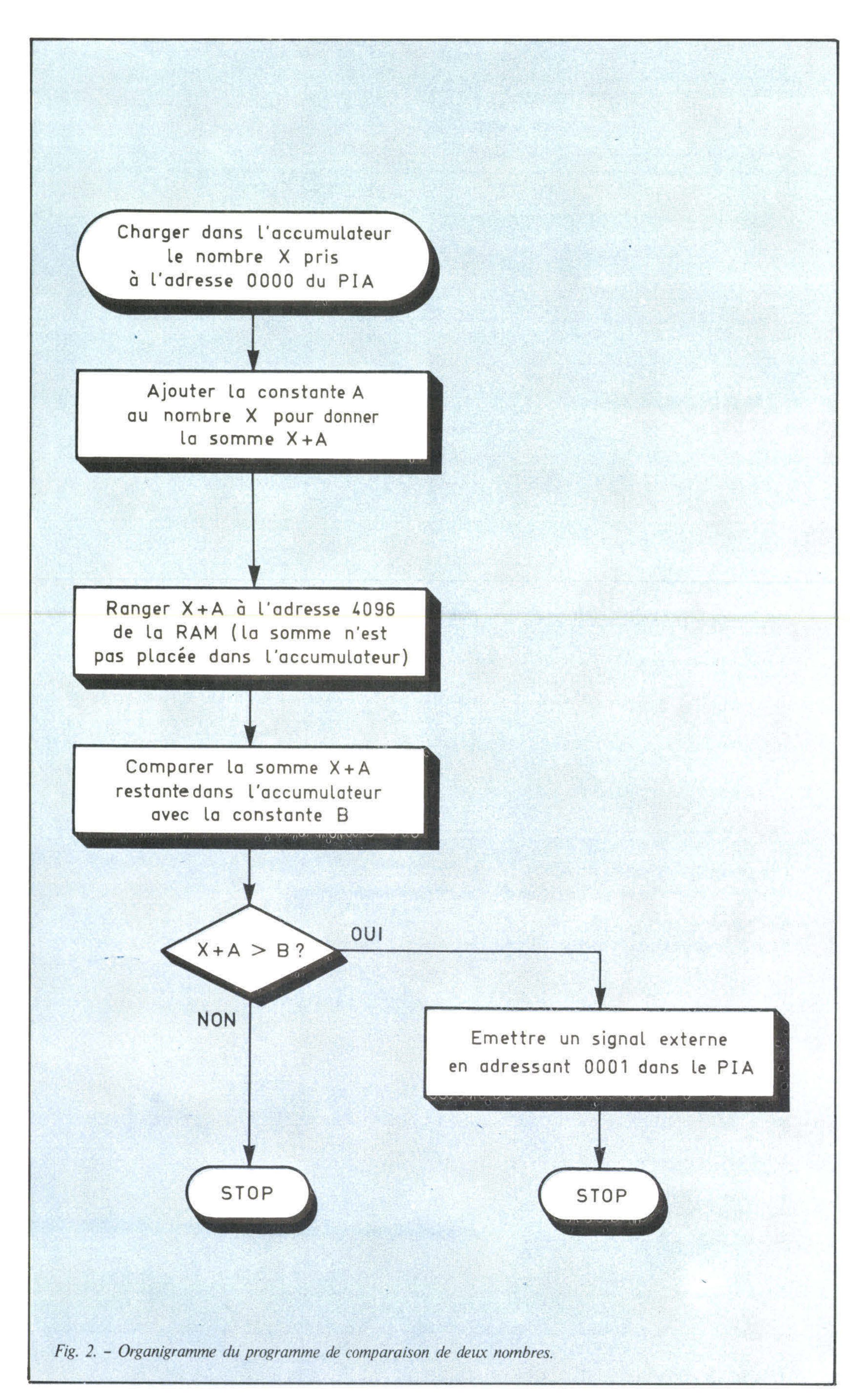


Fig. 2. – Organigramme du programme de comparaison de deux nombres.

Comment le système fonctionne

Notre système étant celui de la **figure 1** qui comprend un microprocesseur, des mémoires vives et mortes et une interface d'adaptation de périphériques (PIA), voyons comment il exécute son programme.

Il faut tout d'abord examiner comment l'information transite dans le microprocesseur. Lorsque le programme a été lancé, le compteur ordinal qui est tenu à jour par la logique de commande, dirige le microprocesseur à chaque étape du programme (comme s'il la montrait du doigt). Sur un plan pratique, l'adresse numérique contenue dans le compteur ordinal est augmentée d'un (incrémentée) et la nouvelle adresse est transférée dans le registre d'adresse. Par exemple, si la dernière instruction exécutée se trouvait à l'adresse 1024, le cycle suivant se déroulera à l'adresse 1025. Quand l'instruction a été trouvée à l'adresse indiquée de la mémoire morte, elle est transmise au décodeur d'instructions par le bus de données. Le décodeur d'instructions est géré par la logique de commande pour diriger les transferts internes de données qu'il s'agisse d'une opération d'addition mettant en jeu l'accumulateur et l'unité arithmétique et logique (ALU), ou d'un simple aiguillage avec incrémentation du compteur ordinal pour passer à l'étape suivante du programme.

Considérons par exemple le programme simple suivant :

– Prendre un nombre codé binaire X à l'extérieur du système par le PIA.

– Ajouter la constante A au nombre X.

– Comparer la somme X + A à une autre constante B.

– Signaler si la somme X + A est plus grande que la constante B.

– Fin de la séquence (STOP).

La **figure 2** est un organigramme, c'est-à-dire un diagramme décrivant l'enchaînement logique des opérations. Le tableau ci-après illustre la disposition des instructions en mémoire morte (ROM).

Pendant la première séquence (1024, 1025 et 1026), le programme utilise l'instruction située à l'adresse 1024 de la ROM pour rechercher la donnée X et ensuite pour l'obtenir du PIA. Pendant ce processus, le registre d'adresse contient la prochaine adresse à examiner qui est incrémentée d'une unité à chaque cycle à condition que le compteur ordinal, qui est toujours incrémenté, soit connecté au registre d'adresse. Le compteur ordinal peut également servir à conserver la trace de la dernière adresse atteinte avant la déconnexion pour fournir l'adresse suivante dans l'ordre numérique lorsqu'il est reconnecté.

Ainsi, pour l'instruction « chargement de l'accumulateur », le compteur ordinal indique 1024 et le registre d'adresse contient la même valeur car l'aiguillage AIG 1 (en position 1) établit une connexion directe. L'instruction de l'adresse 1024 de la ROM est ainsi chargée dans le décodeur par l'intermédiaire du bus de données et de l'aiguillage AIG 2 en position 5.

Le microprocesseur analyse l'instruction et détermine qu'il a besoin de l'adresse de la donnée X. Le contenu de l'adresse 1025 est alors appelé et la valeur numérique 0000 est directement transférée dans le registre d'adresse par le bus de données, l'aiguillage AIG 2, est en position 3, et l'aiguillage AIG 1, en position 2. On notera que l'aiguillage AIG 1 déconnecte le compteur ordinal du registre d'adresse. La donnée X peut maintenant être obtenue directement à l'adresse 0000 du PIA d'où elle est transférée dans l'accumulateur par l'aiguillage AIG 2 en position 4.

La logique de commande amène ensuite AIG 1 en position 1 et AIG 2 en position 5 pour repérer l'acquisition et l'exécution de l'instruction suivante à l'adresse 1026. Le reste du programme est traité d'une manière analogue.

Cette explication est relativement simple car elle concerne un programme court. Cependant, les programmes représentent généralement entre 500 et 10 000 octets et doivent être capables de traiter sans délai des interruptions externes qui peuvent survenir à n'importe quel moment. Il y a en outre plusieurs modes d'adressage qui utilisent le registre d'index et les accumulateurs.

Adresse ROM	Contenu	Type d'information
1024	Chargement de l'accumulateur	Instruction
1025	0000 (adresse du nombre X)	Adresse PIA
1026	Addition constante A à l'accumulateur	Instruction
1027	Constante A (= 2)	Définition de constante
1028	Rangement du contenu de l'accumulateur (X + A)	Instruction
1029	Adresse RAM (4096)	Adresse RAM
1030	Comparaison du contenu de l'accumulateur avec la constante B	Instruction
1031	Constante B (= 2)	Définition de constante
1032	Branchement si X + A > B	Instruction de décision
1033	Adresse de branchement (1035)	Adresse de branchement
1034	STOP	Instruction finale
1035	0001 (adresse du signal)	Adresse PIA
1037	Branchement systématique à 1034	

Tableau I

Implantation physique des instructions du programme en mémoire ROM.

	Compteur ordinal	Registre d'adresse	Aiguillage	
			AIG 1	AIG 2
« Chargement de l'accumulateur »				
Début du chargement	1024	1024	1	
Retour avec instruction	1024	1024		5
Recherche de l'adresse de la donnée X	1025	1025	1	
Retour avec l'adresse de la donnée X	1025	1025	2	3
Recherche de la donnée X à l'adresse 0000	(1026)	0000	2	
Transfert de la donnée X dans l'accumulateur		0000		4
« Addition de la constante A à l'accumulateur »	1026	1026	1	5

Tableau II

Positions des aiguillages et contenus des registres pendant la première séquence du programme (dans le système de la figure 1).

L'assembleur est un programme qui est chargé dans la mémoire du micro-ordinateur pour effectuer l'assemblage d'un programme symbolique.

Création des programmes

Les programmes sont écrits dans des langages qui sont traduits en langage machine (code binaire) par des programmes spéciaux appelés assembleur ou compilateur.

Un programme complet est généralement beaucoup plus long que celui des **figures 3 et 4** et peut atteindre 10 000 octets d'instructions. On imagine combien il serait long et délicat de programmer individuellement ces instructions dans un registre pour les charger en mémoire. On a donc recours à un langage symbolique dont le maniement est beaucoup plus aisé que celui du code binaire et dont les instructions sont convertibles automatiquement en code binaire assimilable par la machine. Ainsi, une instruction d'interruption par programme correspondant au code binaire 00111111 (3F en hexadécimal) s'écrit SWI en langage symbolique (voir **fig. 4**).

Ces instructions symboliques représentées par des codes littéraux mnémotechniques constituent le programme source qui est traduit en code machine binaire par un micro-ordinateur ou un ordinateur muni de l'assembleur correspondant. L'assembleur est un programme qui est chargé dans la mémoire du micro-ordinateur ou de l'ordinateur lorsqu'on veut effectuer l'assemblage d'un programme symbolique.

La **figure 3** illustre un programme symbolique qui est le même que celui du tableau 1 auquel renvoient les commentaires et les adresses figurant sur la partie droite de la page. En fait, le programme a été légèrement modifié pour pouvoir être réellement exécuté sur notre ordinateur. Les étiquettes et les positions de mémoire sont définies avant le lancement du programme et certaines instructions sont combinées. L'assemblage de ce programme source produit un listage complet **(fig. 4)** qui facilite d'éventuelles corrections ou modifications.

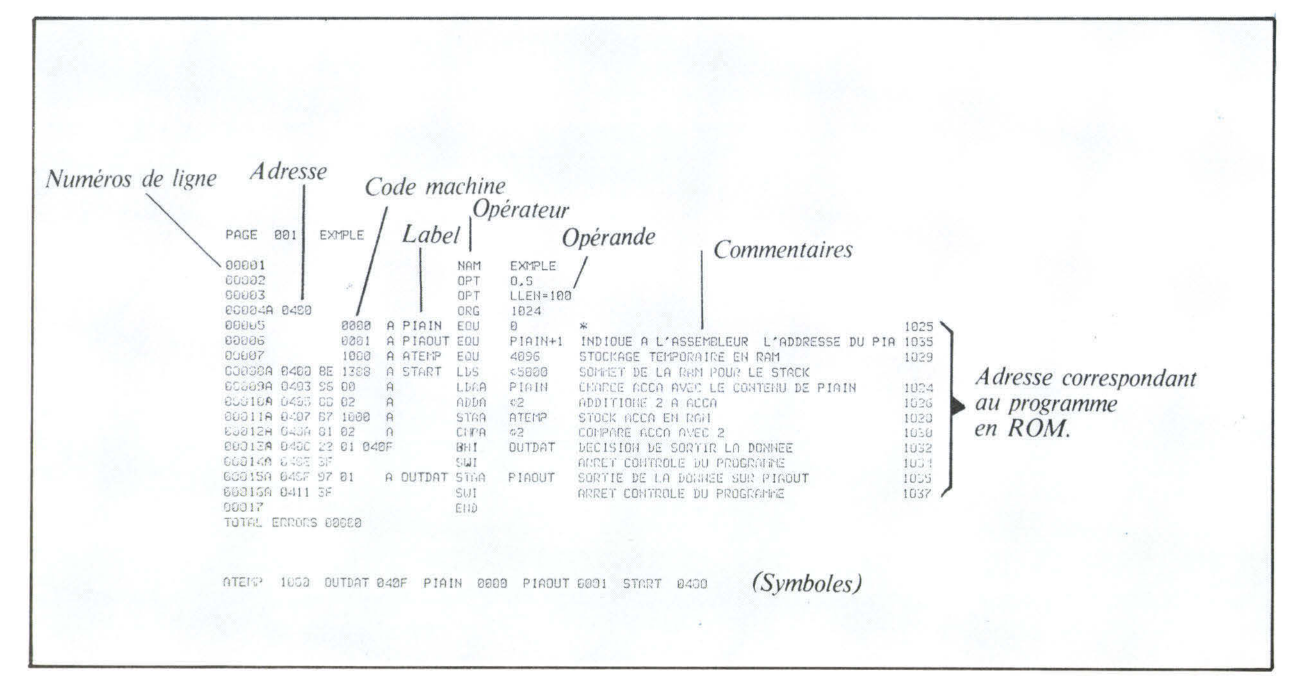
Pendant l'assemblage, l'assembleur adresse à l'opérateur des messages d'erreur signalant les erreurs de syntaxe ou de logique. Pour effectuer les corrections nécessaires, l'opérateur charge un programme spécial appelé **« éditeur »**. Le programme corrigé est ensuite réassemblé et essayé.

L'assemblage manuel des programmes est une tâche fastidieuse que l'on essaie d'automatiser le plus possible au niveau de la machine grâce à des programmes d'assemblage et d'édition intégrés.

Il existe en outre d'autres langages dits « de haut niveau » qui utilisent de courtes phrases se rappro-

```
        NAM    EXMPLE
        OPT    O,S
        OPT    LLEN=100
        ORG    1024
PIAIN   EQU    0         *                                        1025
PIAOUT  EQU    PIAIN+1   INDIQUE A L'ASSEMBLEUR  L'ADDRESSE DU PIA 1035
ATEMP   EQU    4096      STOCKAGE TEMPORAIRE EN RAM               1029
START   LDS    <5000     SOMMET DE LA RAM POUR LE STACK
        LDAA   PIAIN     CHARGE ACCA AVEC LE CONTENU DE PIAIN     1024
        ADDA   <2        ADDITIONE 2 A ACCA                       1026
        STAA   ATEMP     STOCK ACCA EN RAM                        1028
        CMPA   <2        COMPARE ACCA AVEC 2                      1030
        BHI    OUTDAT    DECISION DE SORTIR LA DONNEE             1032
        SWI              ARRET CONTROLE DU PROGRAMME              1034
OUTDAT  STAA   PIAOUT    SORTIE DE LA DONNEE SUR PIAOUT           1035
        SWI              ARRET CONTROLE DU PROGRAMME              1037
        END
```

Fig. 3. – Exemple d'un programme source symbolique destiné à l'assembleur.




```
PAGE  001  EXMPLE

00001                              NAM    EXMPLE
00002                              OPT    O,S
00003                              OPT    LLEN=100
00004A 0400                        ORG    1024
00005            0000  A PIAIN     EQU    0        *                                        1025
00006            0001  A PIAOUT    EQU    PIAIN+1  INDIQUE A L'ASSEMBLEUR  L'ADDRESSE DU PIA 1035
00007            1000  A ATEMP     EQU    4096     STOCKAGE TEMPORAIRE EN RAM               1029
00008A 0400 8E 1388    A START     LDS    <5000    SOMMET DE LA RAM POUR LE STACK
00009A 0403 96 00      A           LDAA   PIAIN    CHARGE ACCA AVEC LE CONTENU DE PIAIN     1024
00010A 0405 CB 02      A           ADDA   <2       ADDITIONE 2 A ACCA                       1026
00011A 0407 B7 1000    A           STAA   ATEMP    STOCK ACCA EN RAM                        1028
00012A 040A 81 02      A           CMPA   <2       COMPARE ACCA AVEC 2                      1030
00013A 040C 22 01 040F             BHI    OUTDAT   DECISION DE SORTIR LA DONNEE             1032
00014A 040E 3F                     SWI             ARRET CONTROLE DU PROGRAMME              1034
00015A 040F 97 01      A OUTDAT    STAA   PIAOUT   SORTIE DE LA DONNEE SUR PIAOUT           1035
00016A 0411 3F                     SWI             ARRET CONTROLE DU PROGRAMME              1037
00017                              END
TOTAL ERRORS 00000

ATEMP  1000  OUTDAT 040F  PIAIN  0000  PIAOUT 0001  START  0400
```

Fig. 4. – Listage complet du programme source assemblé.

chant de la langue (anglaise) parlée. Chaque étape du programme est ainsi plus claire et facile à écrire. Ces langages sont par exemple le **PL1, l'APL,** le **FORTRAN,** le **BASIC,** etc.

Les langages de haut niveau sont utilisés non seulement sur les microprocesseurs, mais aussi sur les ordinateurs de toutes tailles. La traduction en langage machine des programmes écrits en langage de haut niveau nécessite un programme complexe appelé **« compilateur ».**

Il existe des programmes de traduction encore plus élaborés appelés **« interpréteurs »** ou **« macro-assembleurs »** qui permettent de remplacer un grand nombre d'instructions machine par une seule instruction de haut niveau. Avec un macro-assembleur, on peut par exemple écrire directement la macro-instruction « racine carrée de X » qui remplace 25 à 30 instructions machine.

Les assembleurs qui sont conçus pour un autre ordinateur que celui qui utilisera le programme final sont appelés **« cross-assembleurs ».** L'assembleur spécifique du micro-ordinateur auquel est destiné le programme est appelé **« assembleur résident ».** Les mêmes notions sont applicables aux compilateurs qui peuvent être des **« cross-compilateurs »** et aux macro-assembleurs qui peuvent être des **« cross-macro-assembleurs ».**

Les langages de haut niveau ne sont pas toujours utilisables sur un micro-ordinateur car leurs compilateurs ont besoin d'une importante capacité de mémoire et donnent des programmes moins condensés. Par exemple, une fonction directement écrite en code binaire peut représenter 20 instructions machine, alors que la même fonction écrite en langage de haut niveau et compilée peut représenter 35 instructions machine.

Un compromis doit donc être trouvé entre la facilité d'écriture d'un programme et son encombrement en mémoire.

Actuellement, la plupart des utilisateurs de microprocesseurs travaillent en langage machine, mais les langages de haut niveau commencent à se développer grâce à d'importantes baisses sur les prix des mémoires.

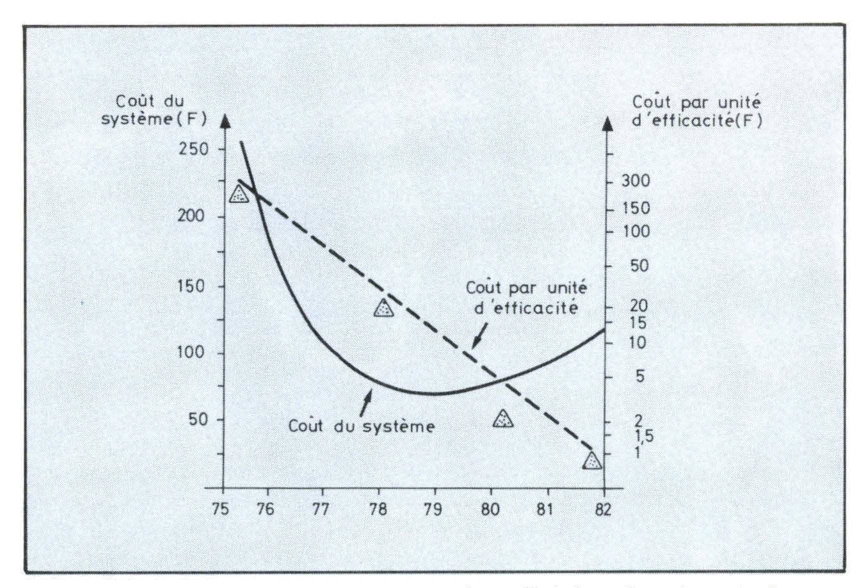


Fig. 5. – Evolution de la relation coût/efficacité d'un système sur la période 1975-1982.

Développement d'une application

Le développement d'une application à base de microprocesseurs nécessite un micro-ordinateur à structure modulaire et logiciel résident.

La décision d'employer des microprocesseurs est conditionnée par les avantages qu'ils offrent sur les solutions mécaniques, électromagnétiques ou électriques en logique câblée. En pratique, l'utilisateur potentiel doit considérer le rapport coût/performances du microprocesseur comparé à ceux des autres solutions.

Ce genre de comparaison a fait l'objet de nombreuses études, mais en règle générale, on peut considérer qu'un produit dont le coût unitaire par série de 1000 dépasse 50 F constitue une application pour laquelle l'emploi des microprocesseurs peut être économiquement justifié.

La puissance de traitement d'un ordinateur se mesure à son **« facteur d'efficacité »** qui est défini arbitrairement comme le nombre d'instructions de son répertoire divisé par la durée d'un cycle complet. Une autre notion utile est le « coût par unité d'efficacité », c'est-à-dire le coût global du système divisé par la valeur du facteur d'efficacité.

La **figure 5** représente l'évolution de la relation coût/efficacité sur la période 1975-1982. Le coût du système (ligne continue) descend vers une valeur minimale au fur et à mesure de l'amélioration du rendement des chaînes de production, mais avec le temps le système devient plus performant et son coût augmente ce qui correspond à la remontée de la courbe.

Il semble assez incroyable que le coût par unité d'efficacité puisse en quelques années tomber de 300 F à 1 F, mais il suffit de se souvenir qu'il y a 30 ans les premiers ordinateurs à tubes à vide effectuaient quatre opérations pour un prix de 5 millions de francs (actuels), ce qui représente un coût par unité d'efficacité de 1,2 million de francs. Il est donc vraisemblable que les perfectionnements constants de la technologie des microprocesseurs puissent aboutir à un coût de 1 F par unité d'efficacité vers 1982. Le temps travaille en faveur des microprocesseurs !

L'emploi d'un microprocesseur dans une application présente deux aspects : le logiciel et le matériel (Hardware). La frontière entre les deux est relativement mal définie car certaines fonctions, comme la multiplication ou le décodage, peuvent se faire soit par hardware, soit par programme. Le choix est conditionné par la capacité de la mémoire et par les moyens de programmation disponibles.

Le coût d'un programme terminé et opérationnel se situe entre 15 et 150 F par octet.

La préparation du logiciel commence par la mise au point d'un organigramme détaillé décrivant toutes les opérations à effectuer. Le programme est ensuite écrit, puis traduit par un ordinateur dans le langage machine du microprocesseur. Après une exécution de contrôle permettant d'éliminer d'éventuelles erreurs, le programme objet est chargé dans une mémoire morte. Cette exécution de contrôle peut se faire sur un ordinateur standard qui possède un programme de simulation du comportement du micro-ordinateur. Pour toutes ces opérations, l'outil le plus souple est cependant un micro-ordinateur spécial de développement ayant un logiciel résident et un adaptateur de périphériques semblables à ceux du système envisagé.

A titre d'exemple, nous pouvons citer le système EXORciser de Motorola. Sa configuration est extrêmement souple car on peut y ajouter à volonté des modules correspondant à des fonctions particulières. Ainsi, la mémoire peut être agrandie par blocs de 2, 8 ou 16 K octets et il existe des cartes d'entrées/sorties, de traitement série/parallèle, etc. Un programme spécial permet de transmettre les ordres au moyen d'un terminal.

En ajoutant un disque souple à l'EXORciser ou à un autre micro-ordinateur similaire, on peut employer des programmes de traduction des langages de haut niveau, tels qu'un compilateur FORTRAN ou un interpréteur BASIC.

Ce genre de micro-ordinateur est un puissant outil de développement d'une application, de la création du logiciel à la simulation du hardware en temps réel.

Par où commencer

Pour de nombreuses applications de l'électronique, le microprocesseur se révèle être une solution souple, puissante et économique, même lorsque personne n'avait jamais pensé à faire appel aux semi-conducteurs pour une telle application.

Lorsqu'on se trouve en face d'une application possible des microprocesseurs, deux questions viennent à l'esprit : où commencer et quel matériel acheter ?

A ce stade, nous pouvons donner quelques conseils simples.

Développement de l'application

On peut diviser le développement en trois phases :

● 1. Réunir suffisamment d'informations sur les microprocesseurs et apprendre à écrire de petits programmes et à les transformer en activités physiques.

S'exercer sur des programmes courts qui seront généralement sans rapport avec l'application envisagée.

Pour cette phase, il suffit d'un système minimal avec une interface élémentaire de communication homme/machine et un logiciel d'appoint relativement simple.

● 2. Création d'un système prototype et de son logiciel utilisables à l'extérieur du laboratoire. Les programmes peuvent être de longueur quelconque, mais doivent pouvoir être rendus pleinement opérationnels dans l'application finale. Cette phase nécessite un système puissant avec des périphériques rapides et la possibilité d'effectuer des simulations en temps réel. Pour faciliter le développement, il est bon de disposer d'aides logicielles et d'un système d'exploitation pour les périphériques.

● 3. Le prototype ainsi créé est reproduit et incorporé à l'application à laquelle il était destiné (de quelques unités à des millions de systèmes).

Ressources financières

Un facteur important est le budget que vous pouvez consacrer au lancement d'un projet de microprocesseurs. De nombreux facteurs ont une incidence sur le coût global, notamment les charges salariales, la formation du personnel, l'habileté des programmeurs, le temps de mise au point des programmes, le niveau local des coûts, etc. Ces variables sont difficiles à évaluer, mais les coûts se montent généralement entre 15 F et 150 F par octet du programme terminé et opérationnel. On peut néanmoins évaluer séparément les investissements nécessaires au système matériel et à son logiciel d'accompagnement.

L'investissement initial peut être classé en trois niveaux :

— Groupe 1 : 1 000 à 10 000 F.
— Groupe 2 : 10 000 à 50 000 F.
— Groupe 3 : plus de 50 000 F.

La **figure 6** indique la phase du développement en fonction de l'investissement et du type d'ordinateur nécessaire. Ce graphique vous permettra de mieux adapter votre budget au stade actuel du développement et de déterminer le type de système dont vous avez besoin.

Bien qu'on ne puisse pas encore trouver un système de développement très puissant pour 1 000 F il est possible d'acquérir à partir d'environ 5 000 F un système extensible dont la puissance pourra être augmentée par l'adjonction de modules supplémentaires sans aucune perte de l'investissement initial. En d'autres termes, on peut se contenter au départ d'un système très rudimentaire dont la complexité croît avec l'évolution du projet. Même si un système complet a été acquis dès le départ, il est possible d'en améliorer l'efficacité en ajoutant des postes de développement et des équipements auxiliaires.

Vous avez aussi la possibilité d'acheter des modules fonctionnels sur cartes de ces systèmes pour les combiner en une machine OEM * incorporée dans vos produits, ce qui évite l'étude d'un hardware spécial.

Enfin, pour évaluer les investissements indirects, vous pouvez vous reporter à la **figure 7** qui donne en pourcentage le temps de création des programmes à l'aide de divers systèmes informatiques. Ce temps représente l'écriture des programmes source, les assemblages et les tests, les corrections et le chargement, mais pas le temps nécessaire à la mise au point qui peut varier considérablement avec la complexité de l'application et l'habileté du concepteur.

Bienvenue dans le monde merveilleux des microprocesseurs ! ■

Daniel QUEYSSAC*

** OEM : Original Equipment Manufacturers.*

** Daniel QUEYSSAC est directeur du Marketing Microprocesseur pour Motorola Europe.*

Production OEM
Développement
Evaluation
Micromodules
_ OEM
_ Cartes
Cartes pour coupes
Terminal Intelligent
Floppy disque
Imprim.te faible coût
Outils de developpement
Exorciser
Intellect.
Intexcept.

Imprim.te rapide
Interface cassette
Systeme de dévelop.t autonome
Terminal faible coût
Test de compo-sants
Kit 2 d'évalua-tion
Kit 1 d'évalua-tion
Carte d'évalua-tion
Carte d'évalua-tion
1000F
10000F
50000F

Fig. 6. – En fonction de la phase du développement et de l'investissement vous pouvez, à l'aide de ce graphique, déterminer le système dont vous aurez besoin.

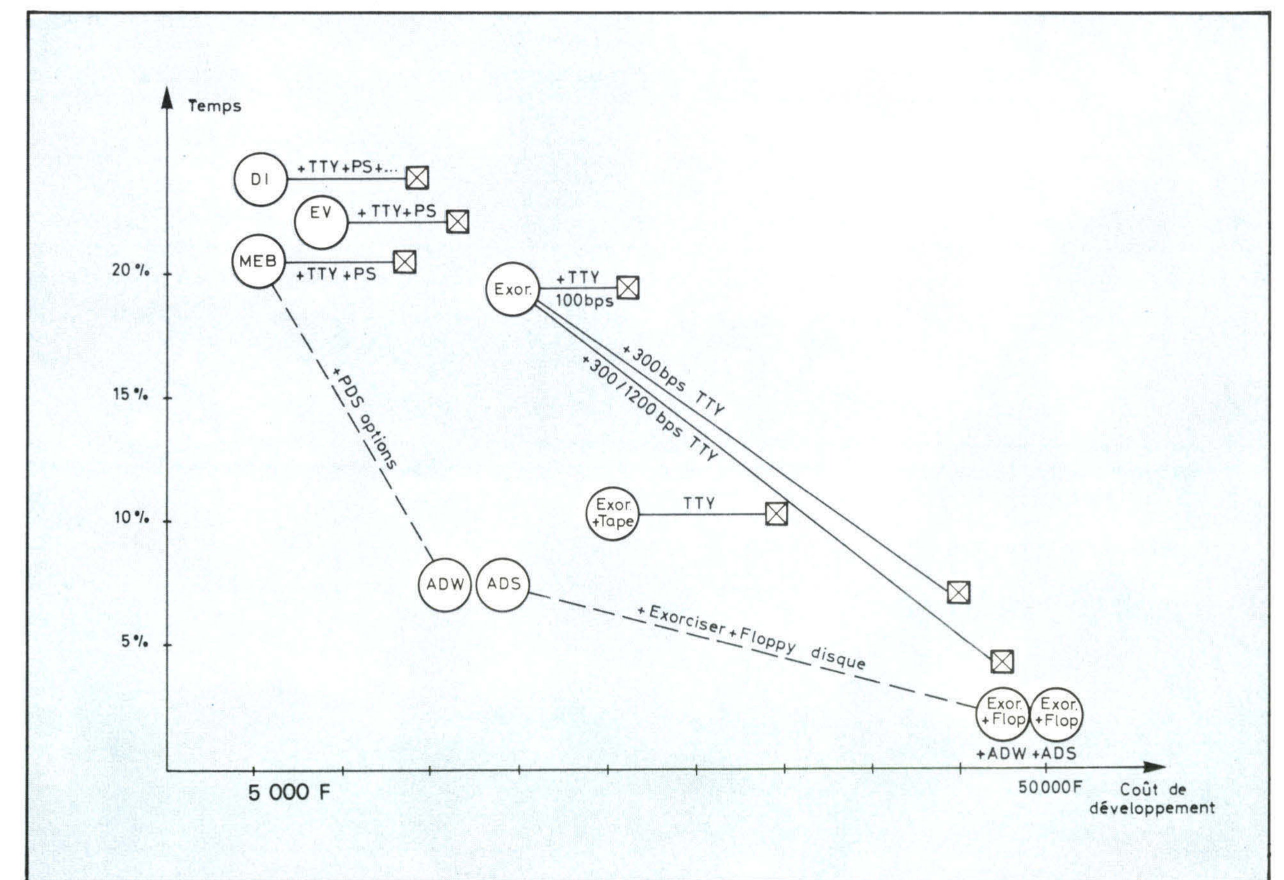


Fig. 7. – Le temps de création des programmes (en %) en fonction des divers systèmes informatiques permet d'évaluer les investissements indirects.

Microcalculateur série 1000

- **Unités centrales : SC/MP** (1 K PROM, 1/2 kram)
  Z 80 (3 K PROM, 1 KRAM), DMA . . .
- **Cartes mémoires** 8 K et 16 K
- **Interfaces cassettes**
- **Interface télétype**
- **Entrées - sorties** industrielles
- Calcul **scientifique**
- Système « **basic** »

PÉRIPHÉRIQUES - SYSTÈMES DE DÉVELOPPEMENT

Distribué par :

— DEBELLE, 13, rue Baptiste-Marcet, Z.I. Fontaine Sassenage, 38600 Fontaine.
— FACEN LILLE, 6, rue Emile-Rouzé, 59000 Lille.
— FACEN NANCY, Z.I. D'Heillecourt, 54140 Heillecourt.
— FACEN ROUEN, boulevard Industriel, 76800 Saint-Etienne-du-Rouvray.
— FACEN STRASBOURG, Z.I. rue Vauban, 67450 Mundolsheim.
— FENNER GENEVE, 18, rue de Miremont, 1211 Genève 25.
— GENERIM, avenue de la Baltique, Z.A. de Courtabœuf, B.P. 88, 91403 Orsay.
— PENTASONIC, 5, rue Maurice-Bourdet, 75016 Paris.
— R.T.F., 73, avenue Charles-de-Gaulle, 92202 Neuilly-sur-Seine.

Ventes par correspondance : Notices et tarifs sur simple demande.

« Points micro »

— 185 av. de Choisy, 75013 Paris
— 5 rue Maurice Bourdet, 75016 Paris
— 9 bis rue du Bas-Chamfleur, 63000 Clermont Fd
— 6 rue de la Loi, Mulhouse
— 32 rue Oberlin, Strasbourg
— 13 rue Baptiste Marcel, 38600 Fontaine

APPLE II

LE MICRO-ORDINATEUR SIMPLE, PERFORMANT ET COMPLET





ILLEL CENTER

143, AVENUE FÉLIX FAURE - PARIS 15e
TÉL. : 554.83.81

DEMONSTRATION LE SAMEDI APRES-MIDI OU SUR RENDEZ-VOUS

MICRO INFORMATIQUE APPLE

I – MICRO-ORDINATEURS et FLOPPY DISKS APPLE II

APPLE II est un micro-ordinateur complet, assemblé et testé; Il est livré avec câbles de raccordement, manuel d'utilisation, cassettes de démonstration, ainsi que deux manettes pour tracés de diagrammes.

Configuration

16 K .... F 9.800 HT
32 K .... F 12.200 HT
48 K .... F 14.600 HT

Vous pouvez nous consulter pour des configurations différentes.

MINI FLOPPY "DISK II"

D'une capacité de 116 octets formatés, d'un accès direct rapide; l'ensemble comprenant le Driver, le Contrôleur et le Disc Operating System (D.O.S.) est livré avec manuel et deux disquettes .... F 4.950 HT

II – INTERFACES

- Interface de codage couleur RVB (nécessite une prise RVB sur le téléviseur) .... F 780 HT
- Interface de codage couleur SECAM .... F 1280 HT
- Modulateur noir et blanc seul F 280 HT
- Interface de sortie parallèle pour imprimante ou applications diverses F 1280 HT
- Interface de série en mode V 24, RS 232 C (vitesse commutable de 75 à 19.200 bauds permettant le raccordement à tous les périphériques ou systèmes au standard de communication RS 232 C) .... F 1280 HT
- Interface de communication vitesse de 110 à 300 bauds en RS 232 C (destiné plus particulièrement aux liaisons par modem et à l'utilisation de l'APPLE comme terminal) .... F 1280 HT
- Interface de reconnaissance de la parole, permet de reconnaître un mot parmi un vocabulaire programmable de 32 mots (reconnaissance par analyse heuristique) .... F 1700 HT
- Interfaces spéciales ...... suivant devis

III – PÉRIPHÉRIQUES

IMPRIMANTES : Economiques sur papier métallisé.
AXIOM 801 (20, 40 ou 80 caractères) .... F 4800 HT
AXIOM 820 (printer et plotter permettant graphismes) .... F 5500 HT
Sur papier ordinaire :
CENTRONICS 779 .... F 8200 HT
DATA 100 type S 1200 ...... F 15.950 HT

IV – LOGICIELS DE DÉMONSTRATION (SOFTWARE)

Ces programmes sont disponibles en support cassettes.

Gestion de stock .... F 300 HT
Gestion de compte en banque F 300 HT
Comptabilité générale ...... F 600 HT
Jeux divers I ou II, l'unité ..... F 100 HT
Affichage publicitaire .... F 100 HT

Ils peuvent être également livrés selon la demande sur supports disquettes, (prix suivant devis).

V – FOURNITURES ET DOCUMENTATION

Disquettes, l'unité .... F 35 HT
Housse en vynil .... F 300 HT
Mémoires additionnelles : suivant devis
Reference manual (anglais) .. F 100 HT
Reference manual (français) . F 150 HT
Manual BASIC (anglais) ...... F 100 HT
Manual BASIC (français) ..... F 150 HT
Applesoft (anglais) .... F 100 HT
Applesoft (français) .... F 150 HT

Les prix des modèles présentés peuvent être modifiés sans préavis, et ne sont donnés qu'à titre indicatif.

Les ordinateurs APPLE II et les mini Floppys APPLE II sont garantis six mois à dater de la livraison, contre tout vice de fabrication.

Des contrats (pièces et main d'œuvre) de maintenance peuvent être souscrits au-delà de cette période.

Les autres périphériques bénéficient de la garantie spécifique de chaque fabricant.

ATTENTION : Les prix cités sur le présent tarif étant hors taxes, il y a lieu de les majorer de 17,60%.

DU MICROPROCESSEUR AU MICRO-ORDINATEUR

Séminaire exceptionnel 98 F TTC

PARIS - 25 NOVEMBRE 78
Maison de la Chimie - 28, rue St-Dominique - Paris 7e

A l'occasion de son lancement, la revue MICRO-SYSTÈMES organise avec la collaboration de SYBEX une journée de formation et d'information sur l'utilisation des microprocesseurs dans la conception des micro-ordinateurs à usage personnel.

I - Initiation aux microprocesseurs (9 h-12 h)

Introduction - Fonctionnement d'un microprocesseur - Composants d'un système - Les familles de microprocesseurs - Applications - Programmation d'un microprocesseur - Développement d'un produit - Aides au développement - Sélection d'un microprocesseur - Evolution et Perspectives.

II - Les micro-ordinateurs (14 h-16 h)

Définition et constitution d'un micro-ordinateur - Fonctionnement - Applications - Historique - Les périphériques - Imprimante, disque, écran, magnétophone - Les différents types de micro-ordinateurs : leur choix, leurs applications - L'avenir.

Conférencier : Rodnay Zaks : ingénieur E.C.D., Ph. D. Univ. de Berkeley

est l'auteur de plus de 30 ouvrages sur les microprocesseurs dont le dernier, « An introduction to personal and business computing », best-seller aux USA, est désormais disponible en français.

III - Débat (16 h-18 h)

— Comment choisir son micro-ordinateur à usage personnel.
— Tribune libre : les différents constructeurs et distributeurs défendront leur produit face à l'assistance (président de Session : prof. David).

IV - Exposition (entrée libre)

Durant tout le séminaire une exposition permanente présentera les principaux modèles de micro-ordinateurs en vente sur le marché français.
Liste des produits présentés : REA avec le AIM 65 / JCS avec le MK 14 et le Nascom 1 / Heathkit avec le H 8 et le H 11 / Proteus avec le Proteus III et IV / ERN avec le VIM / Euro-Computer Shop avec le SORCERER / Mato avec le TRS-80 / Sonotec avec Apple-II / PROCEP / JANAL Computer Shop / IVS, etc.



EUROPE :
313, rue Lecourbe
75015 PARIS
Tél. : (1) 828.25.02
Télex : 200858

USA :
2020 Milvia St,
Berkeley, CA 94704
Tél. : (415) 848.82.33

MICRO SYSTEMES

15, rue de la Paix, 75002 Paris. Tél. : 296.46.97

COUPON D'INSCRIPTION

☐ Inscrivez-moi à la journée micro-ordinateurs
☐ Envoyez-moi tous renseignements sur les livres SYBEX
☐ Ci-joint 98 F T.T.C. à l'ordre de Micro-Systèmes

Nom ______________ Profession ______

Société ______________

Adresse ______________

______________ Tél. ______

ENVOYEZ A : MICRO-SYSTEMES, 15, rue de la Paix
75002 Paris

M.S 11.78


MICRO

Microprocesseur équivalent à un minicalculateur

Le microprocesseur 9440 Microflamme™ est le premier microprocesseur bipolaire 16-bits apparu sur le marché capable d'exécuter un jeu d'instructions avec les performances du mini-calculateur.

Le dispositif 9440 rassemble sur une seule pastille de silicium l'unité arithmétique complète d'un mini-calculateur, l'ensemble étant présenté dans un boîtier de 40 broches. Les applications principales de ce nouveau dispositif sont : traitement d'informations sous un grand nombre de formes tel que le contrôle de processus, l'instrumentation, les télécommunications, les centraux PBX et PABX ainsi que tout traitement du type multi-processing et terminaux intelligents.

Logiciels

Le support logiciel disponible : FIRE™ (Fairchild Integrated Real Time Executive) comprenant des outils conversationnels de mise au point tels que chargeur binaire, sauvegarde et restauration de programmes.

De plus, le microprocesseur Microflamme est capable d'exécuter l'ensemble des instructions du mini-calculateur Data General NOVA 1200.

Le logiciel "FIRE" comprend un **éditeur de texte** permettant la mise au point aisée des programmes symboliques et un BASIC résident en PROM disponible dès maintenant.

Technologie avancée

Le nouveau microprocesseur a été conçu à partir d'une technologie très évoluée appelée I^3L (Isoplane Integrated Injection). Cette technologie combine les avantages de la vitesse des circuits bipolaires avec la densité et la faible dissipation de la technologie MOS.

Le chip 9440 comprend également des circuits d'entrée et de sortie compatibles TTL qui assurent un interface aisé avec les autres circuits logiques, les mémoires PROM et RAM.

Kit 9440

Afin de vous familiariser avec ce microprocesseur, Fairchild propose un kit économique comprenant l'élément 9440, 4K mots de 16 bits en mémoire dynamique, les composants SSI et MSI ainsi que les manuels d'instructions FIRE 1.

Ce kit permettra au niveau d'une carte d'effectuer des opérations de base à l'aide de ce dispositif performant.

Nouveaux logiciels

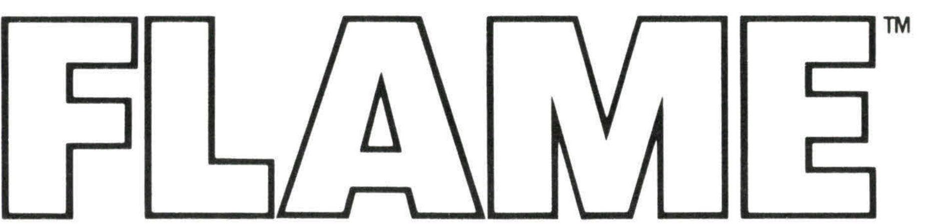
Des logiciels FIRE plus élaborés ainsi que d'autres cartes de développement seront rendus disponibles au cours des prochains mois.

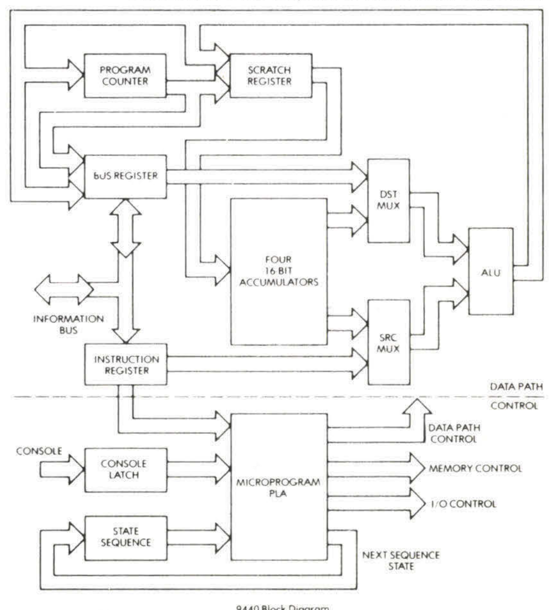
Le nouveau logiciel comprendra un système floppy disque avec un DOS (disc operating system).

Les compilateurs FORTRAN et PASCAL seront disponibles début 79.

De nouveaux circuits seront également disponibles tels que mémoires dynamiques TTL 16K, une unité de contrôle mémoire à accès direct (D.M.A.), des circuits d'entrée/sortie puissants et les opérations de multiplication et division câblées.

NOVA est une marque déposée de Data General Corporation





9440 Block Diagram

FAIRCHILD

Pour toute information concernant l'ensemble de la famille 9440, veuillez nous contacter afin d'obtenir la brochure correspondante.

Tél. : 584.55.66

MICRO SYSTEMES

Comment souscrire un abonnement ?

— **par correspondance,** en utilisant le bulletin d'abonnement ci-dessous (ou une photocopie), à retourner à : MICRO-SYSTEMES, 2 à 12, rue de Bellevue, 75940 Paris Cedex 19. Tél. : 200.33.05.

— **chez votre marchand de journaux habituel,** en lui remettant le bulletin d'abonnement ci-dessous dûment rempli.

A découper suivant le pointillé

Bulletin d'abonnement

Nos tarifs :

	FRANCE	ETRANGER
MICRO SYSTEMES (6 numéros) bimestriel - 1 an	(1) ☐ 45,00 F	70,00 F

Informations :

— **pour les changements d'adresse :** joindre la dernière étiquette d'envoi, ou à défaut, l'ancienne adresse accompagnée de la somme de 2,40 F en timbres-poste, et des références complètes de la nouvelle adresse.

— **pour tous renseignements** ou réclamations, concernant votre abonnement, joindre la dernière étiquette d'envoi.

(1)

☐ Je m'abonne pour la première fois à partir du numéro paraissant au mois de :

☐ Je renouvelle mon abonnement : et je joins ma dernière étiquette d'envoi.

Je joins à ce bulletin la somme de : . . . . F
par :
— chèque postal ☐ sans numéro de compte
— chèque bancaire ☐
— par mandat-lettre ☐
A l'ordre de **MICRO SYSTEMES**

(1) ☐ Mettre une croix dans les cases ci-dessus correspondantes.

Ecrire en capitales, n'inscrire qu'une lettre par case. Laisser une case entre 2 mots. Merci

Nom, Prénom (Attention : prière d'indiquer en premier lieu le nom suivi du prénom)

Complément d'adresse (Résidence, Chez M., Bâtiment, Escalier, etc.)

N° et Rue ou Lieu-Dit

Code Postal *Ville*

Dept	Cne	Quartier

13 *19*

Ne rien inscrire dans ces cases

Le dessin de MICKEY

Comment dessiner sur un ordinateur ?

Il y a deux manières de dessiner sur un ordinateur :
- par le calcul,
- par la copie point par point du programme.

Dans le premier cas, l'ordinateur calcule réellement la position de chaque point du dessin. Il doit donc résoudre une expression ou une équation plus ou moins compliquée suivant la figure à réaliser. Le résultat de chacun de ces calculs produit un déplacement de la tête d'impression généralement égal à l'entier des nombres obtenus. La frappe elle-même pourra être « variable » car un caractère est représenté par un octet hexadécimal ou une quantité décimale variable.

La façon de dessiner Mickey dans ce programme est beaucoup plus simple puisque le modèle de Walt Disney est représenté directement dans le, programme sous forme d'instruction « PRINT » (ce qui vous permettra de bien vous familiariser avec cette instruction).

Ainsi, chacun peut dessiner de la même manière de nombreuses figures moyennant un peu de talent.

Nos lecteurs qui désirent reproduire exactement ce dessin peuvent copier intégralement le programme ligne par ligne.

Mickey apparaît dans ce cas de deux façons : dans le programme lui-même, puisque ce sont les instructions PRINT que l'ordinateur exécute mais les instructions et les guillemets sont reproduits sur le listing ; en lançant le programme par le « RUN » habituel, on obtient la même image sans les guillemets avec éventuellement les commentaires s'ils existent.

Bon amusement... ■

```
>
10 REM ***MICKEY
200 PRINT "                    OKKKKK80"
201 PRINT "                 0000000000000000"
202 PRINT "                000000000000F0F0000"
203 PRINT "                000000ROROROROR RR00"
204 PRINT "                 00000000RRORORORO"
205 PRINT "                   000000000000RO"
206 PRINT "                      0000000"
207 PRINT "                  0000000000000000"
208 PRINT "                0000000000000000000000"
209 PRINT "              0000000000000000000000000000"
210 PRINT "            00000000000000000000000000000000"
211 PRINT "     00     00000000000000000     000    000 "
212 PRINT "   00000   00000000000000000       0       00 "
213 PRINT "  0000000  000000000000000000  . .  `0  . .   0        0000"
214 PRINT "  0000000  000000000000000000  00  0  00   0        000000"
215 PRINT "  0000000000000000000000000000 00. 0000  000000   000000 "
216 PRINT "  00000000000000EOEOEOEOE000000000000000000000000000000000"
217 PRINT "   000000   000000000000000000000000000000000000000000"
218 PRINT "    0000     000000000000000000000000000000000000000000"
219 PRINT "              00000000000F0F0F0F0F0F0F0F0F0F00000000000"
220 PRINT "                000000000D0D0D0D0D0D0D0D000000000000000"
221 PRINT "                   0000000PPPPIIIII000000PPP00000000"
222 PRINT "                       00000000IIIII000000000000"
223 PRINT "                          000000000000000000"
224 PRINT "                                000000"
225 PRINT "                                000000"
226 PRINT "                        TT       TTTTTT       TT"
227 PRINT "                        TTTT     TTTTTT     TTTT"
228 PRINT "                        TTTTTTTTTTTTTTTTTTTTTTTT"
229 PRINT "                        TTTTTT           TTTTTTT"
230 PRINT "                        TT       TTTTTT       TT"
231 PRINT ""
232 PRINT ""
233 PRINT ""
235 PRINT "DESSINE PAR C. R. ISMAN SUR UN MICRO-ORDINATEUR MCZ-1/05"
237 PRINT "SUPERVISE PAR LAURENCE (ZILOG-A2M), JEAN-MARC, DORIS, GILLES"
238 PRINT "APPROUVE PAR MICROSYSTEMES"
239 PRINT "D'APRES L'OEUVRE DE WALT DISNEY"
```

```
DESSINE PAR C R ISMAN SUR UN MICRO-ORDINATEUR MCZ-1/05
SUPERVISE PAR LAURENCE (ZILOG-A2M), JEAN-MARC, DORIS, GILLES
APPROUVE PAR MICROSYSTEMES
D'APRES L'OEUVRE DE WALT DISNEY
```

Le dessin de Mickey apparaît dans le programme lui-même avec les instructions REM, PRINT et les guillemets. Après exécution du programme, il est imprimé avec les commentaires qui l'accompagne (en médaillon).

Une journée exceptionnelle « Micro-ordinateur »

A l'occasion de son lancement, MICRO-SYSTEMES organise avec SYBEX, pour la première fois en France, une journée exceptionnelle « Micro-ordinateur ».

Journée d'information et de formation sur les microprocesseurs et sur la conception et l'utilisation des micro-ordinateurs à usage personnel, cette manifestation sera marquée par deux événements d'importance. En effet, la première exposition de micro-ordinateurs personnels ouvrira ses portes et simultanément se tiendra un séminaire réservé à tous ceux qui, sans être nécessairement des professionnels, n'en sont pas moins des amateurs enthousiastes.

L'exposition *(entrée libre)*

Cette exposition sera entièrement consacrée aux micro-ordinateurs à usage personnel.

L'exposition à deux buts : vous permettre d'avoir une vue d'ensemble des systèmes micro-ordinateurs disponibles et commercialisés actuellement en France et vous offrir la possibilité d'entrer en contact avec les distributeurs, revendeurs et fabricants de ces produits.

Que votre budget soit de 1 000 F ou de 10 000 F, que vous recherchiez de nouvelles idées de jeux, simulation ou un système de gestion, cette exposition vous apportera nécessairement une réponse.

Il se peut aussi que vous souhaitiez tout simplement vous instruire et vous faire une opinion sur cette informatique qui, nouvellement, a su trouver une dimension humaine.

Cette manifestation est une excellente occasion de venir faire le point sur l'avancement et le développement de la micro-informatique. Ainsi, vous pourrez comparer l'évolution suivie par les deux grandes tendances que sont les systèmes « clés en main », tels que le Proteus III, le PET, l'APPLE II, et les systèmes en kit comme le H 8 ou le Nascom I.

En avant première, nous vous invitons à parcourir avec nous les stands de cette exposition « Micro-ordinateur », première du nom, ouvert aux non professionnels.

Les Exposants

E.M.R.

s'est fixée dès son origine un double objectif en visant à la fois le marché industriel et le marché amateur.

S'appuyant sur un module de sa conception, cette société propose un système versatile dont les trois axes de développement sont : l'automatisme, la petite gestion et les transmissions. De ce fait, ce système peut aussi bien s'employer dans les automatismes industriels, dans des **applications pour radio amateurs** que dans la réalisation de jeux.

E.N.R.

se consacre essentiellement à l'importation et à la distribution de composants électroniques. En matière de micro-informatique, cette société représente les marques Synertek, Solid State Scientific et Data Board.

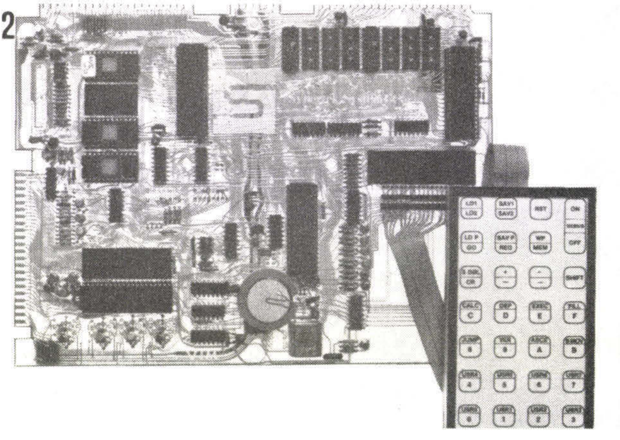
Actuellement, Synertek propose sur le marché français la carte microcalculateur VIM.1.

EURO-COMPUTER-SHOP

Spécialisée dans la commercialisation des micro-ordinateurs et périphériques associés, cette société est à l'origine du premier magasin d'ordinateurs ouvert en France.

La gamme de ses produits s'adresse aussi bien aux petites et moyennes entreprises qu'aux amateurs. Parmi les principaux systèmes commercialisés par cette société, nous pouvons citer les micro-ordinateurs Sorcerer et Horizon, tous deux construits autour du Z-80.





1 - Cartes et micro-ordinateurs EMR conçus autour du SC/MP de National Semiconductor.

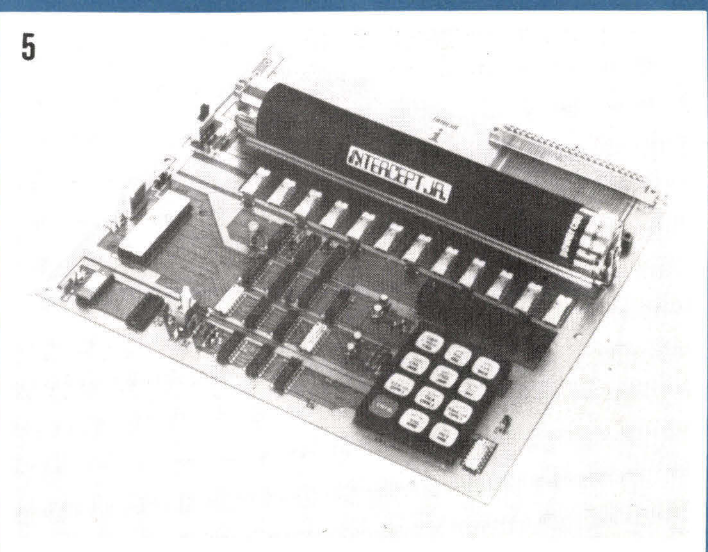
2 - Micro-calculateur VIM 1 distribué par la société ENR.

3

4



5



3 - Le micro-ordinateur Sorcerer commercialisé par Euro Computer Shop est construit autour du Z 80.

4 - Heathkit a conçu pour les amateurs un micro-ordinateur en kit référencé H 8.

5 - L'intercept Junior de Intersil.

HEATHKIT

le plus grand fabricant mondial de kits électroniques, a introduit sur le marché sa nouvelle gamme de mini-ordinateurs et périphériques.

Qu'il s'agisse du H 8, ordinateur de 8 bits, ou du H 11, ordinateur de 16 bits, ces matériels s'adressent indifféremment à l'informaticien amateur ou à la petite entreprise.

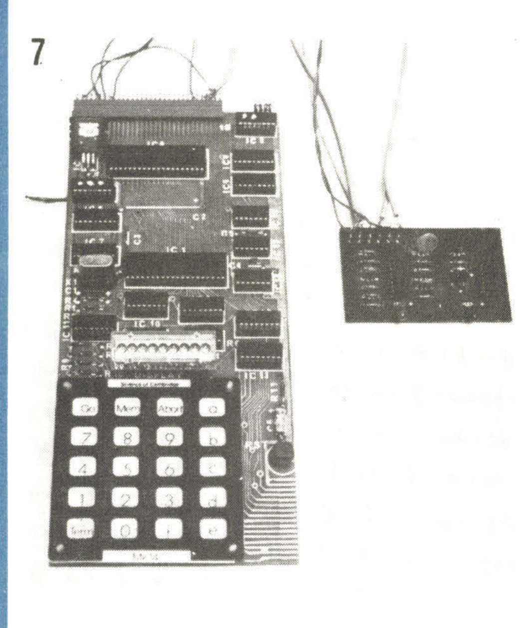
INTERSIL

lançait en 1975 une famille de microprocesseurs en technologie C.MOS bâtie autour de l'IM-6100. Leader incontesté dans ce domaine, cette société propose en outre un système d'introduction aux microprocesseurs : l' « Intercept-Junior » ainsi qu'un kit baptisé « Sampler ».

6



7



8

JANAL-COMPUTER-SHOP

a pour vocation la distribution de systèmes allant des cartes d'initiation aux micro-ordinateurs évolués tels que le PET ou l'APPLE II.

J.C.S. COMPOSANTS

assure la commercialisation en France de deux nouveaux micro-ordinateurs qui sont le MK 14 et le Nascom I.

TANDY-RADIO-SHACK

dispose d'une longue expérience dans le domaine de l'électronique. Nous lui devons le TRS-80 qui, équipé d'un microprocesseur Z-80, constitue un micro-ordinateur doté d'une grande souplesse d'exploitation et utilisable en Basic. Ce matériel est exclusivement diffusé pour Paris et la région parisienne par MATO-SON.

PROCEP

commercialise en France les produits de Commodore et de MOS Technology. Cette société présente le PET, micro-ordinateur complet avec Basic résident, ainsi qu'un des micro-ordinateurs sur carte les plus connus : le KIM-1.

PROTEUS S-A

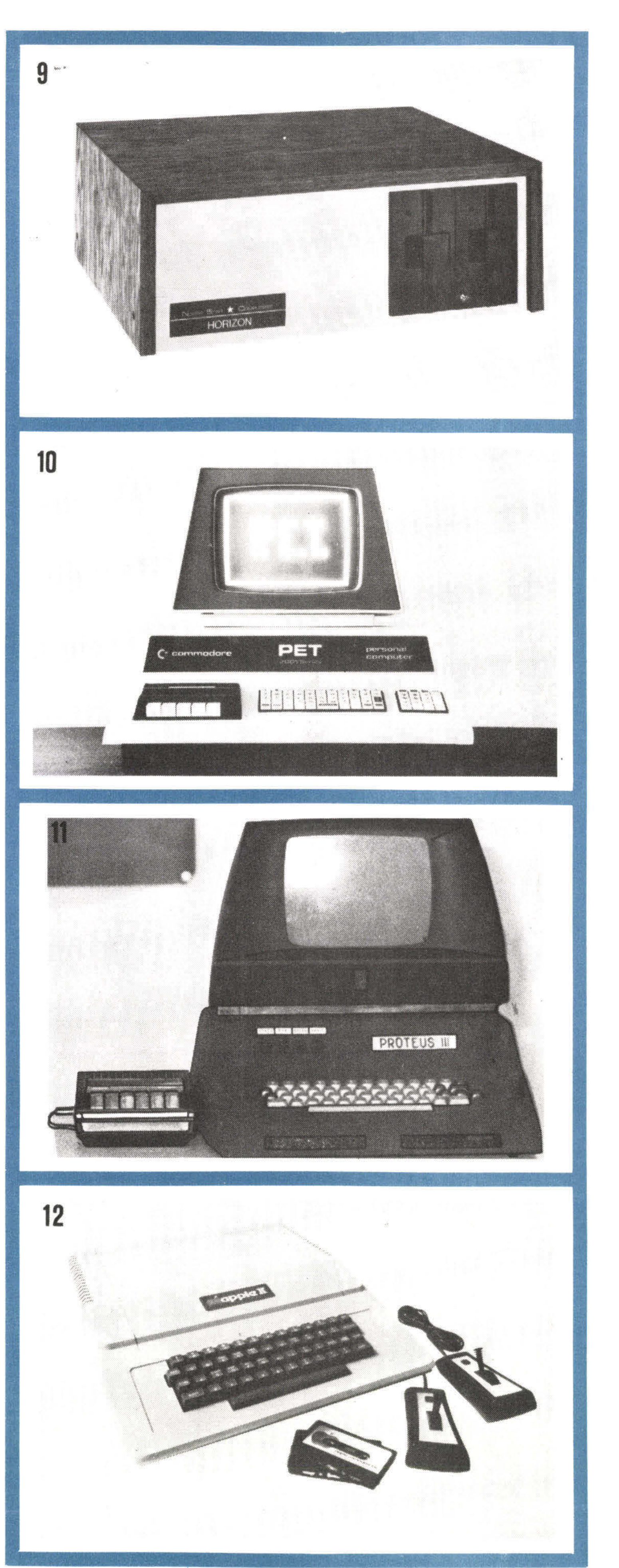
Société française du groupe Proteus International, consacre ses efforts à la fabrication et à la vente, en France et à l'étranger, d'une gamme complète de systèmes micro-informatique. Actuellement la série Proteus III constitue un micro-ordinateur en Basic étendu, orienté vers les problèmes de gestion.

R.E.A.

assure dans son département microprocesseurs la vente des produits de chez Rockwell, R.C.A. avec respectivement pour micro-ordinateurs, la carte KIM-1 et le kit de développement Cosmac-VIP. Commercialise depuis peu le nouveau micro-ordinateur AIM-65 de Rockwell.

SONOTEC

est l'importateur des micro-ordinateurs produits par la société californienne Apple-Computer dont le modèle le plus apprécié en France est l'Apple II.



6 - NASCOM 1 de JCS Composant.

7 - Le MK 14 est un kit d'initiation bâti autour du SC/MP de NS.

8 - REA commercialise depuis peu le nouveau micro-ordinateur AIM 65 de Rockwell.

9 - Floppy Disk de l'ordinateur Horizon de North Star Computers.

10 - Un micro-ordinateur complet : le P.E.T.

11 - Le nouveau micro-ordinateur Proteus III de Proteus International.

12 - Apple II : distribué par Sonotec.

Le Séminaire

De par les puissances de calcul qu'offrent les microprocesseurs, ces composants électroniques engendrent une révolution dans notre monde pétri de technique.

L'électronique est en pleine mutation, l'électronique programmée est née, l'ère de l'ordinateur personnel vient de commencer.

A l'aube de cette nouvelle étape, il est bon que chacun puisse venir s'informer, s'initier à ces nouvelles techniques et rencontrer des personnes qui comme vous partagent la même passion. C'est pour répondre à ce besoin que ce séminaire a été organisé.

Pour un droit de participation que nous avons voulu le plus bas possible, cette journée vous permettra d'assister à une initiation aux microprocesseurs et micro-ordinateurs présentée par l'un des grands spécialistes mondiaux en la matière : M. Rodnay Zaks, directeur de Sybex-Europe, homme au passé universitaire prestigieux, mais récent.

Auteur de plus de trente ouvrages sur les microprocesseurs, son dernier titre, « Introduction aux micro-ordinateurs à usage personnel et commercial » fait figure de best-seller actuellement aux U.S.A.

Récemment sur Antenne 2, dans l'émission « Question de temps », du 13 septembre, consacrée au phénomène hobbyiste américain, il nous a fait découvrir la place que pouvait prendre le microordinateur dans notre environnement quotidien.

Cette émission avait été, en grande partie, tournée à Sybex U.S.A. La présence à ce séminaire n'exige pas de grandes connaissances techniques, bien qu'une expérience en électronique ou informatique constitue une aide appréciable.

En fait, il s'adresse à ceux qui ont pris conscience du phénomène microprocesseurs et qui souhaitent acquérir rapidement les bases nécessaires à la compréhension et à l'utilisation des micro-ordinateurs.

La conclusion de ce séminaire sera un débat au cours duquel différents constructeurs et distributeurs du marché français viendront présenter et défendre leur matériel.

Le Programme

LA MATINÉE (de 9 h à 12 h)

sera consacrée à une initiation aux microprocesseurs permettant ainsi à chacun de se familiariser avec cet élément avant même d'aborder le thème central de la journée que constitue le micro-ordinateur à usage personnel.

Loin d'être sommaire, cette première intervention de Rodnay Zaks retracera la récente, mais ô combien riche, histoire du microprocesseur.

Ainsi, partant des techniques d'intégration appliquées aux circuits digitaux, le microprocesseur apparaîtra dans sa génèse pour être étudié dans sa structure interne puis dans son mode de fonctionnement.

C'est en s'appuyant sur cet indispensable préliminaire que l'on pourra percevoir de quel outil l'on dispose. Fort de cet acquis, il devient possible d'imaginer et de concevoir des applications intégrant un ou plusieurs de ces composants programmables.

En fait, il reste à envisager suivant quelles étapes et quels moyens il est permis de transformer une spécification décrivant une application donnée en une suite d'instructions exécutables par le microprocesseur : ce qui, en terme simple, s'appelle programmation.

Parmi les nombreux microprocesseurs présents sur le marché, il est évident que dans le cadre d'une application déterminée, un choix s'impose, car tous ne sont pas équivalents.

Si, pour chacun d'eux il semble possible de définir un domaine d'application, il demeure nécessaire de retenir quelques critères de sélection qui peuvent se résumer à : puissance de calcul, simplicité de mise en œuvre, prix de revient et bien entendu disponibilité aussi bien en ce qui concerne le composant lui-même que de son logiciel.

L'APRÈS-MIDI (de 14 h à 16 h)

se poursuivra par la présentation des micro-ordinateurs, toujours sous la conduite de notre brillant conférencier de la matinée.

Si, pour l'homme de la rue ce terme évoque quelques idées imprécises reposant sur des applications anecdotiques largement répandues par une presse qui, dès le début de ce phénomène, a contribué de façon appréciable à sa diffusion dans notre pays, on ne peut en rester là.

Il est intéressant à ce niveau, de voir comment se définit et par là-même comment se constitue un tel système.

Dans cette approche, nous verrons comment un microprocesseur permet, moyennant la mise en place de quelques composants supplémentaires, l'esquisse d'une architecture logique dont vous découvrirez les règles qui régissent la construction de cet édifice.

Véritable clef de voûte de ce système, le microprocesseur en est l'unité centrale, conçue pour travailler suivant un programme stocké dans une mémoire...

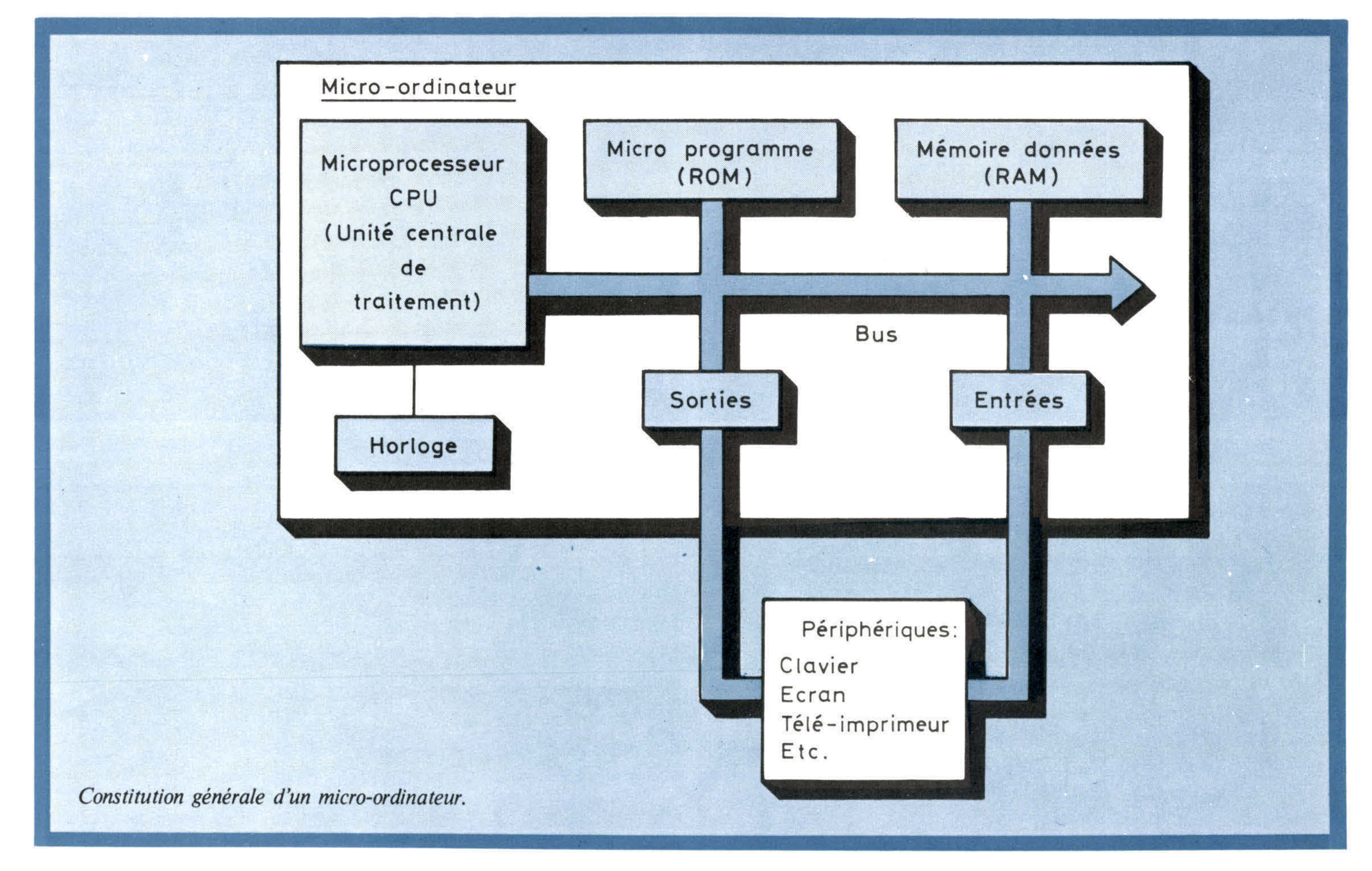
Sous la conduite de ce programme, notre système traitera les données ainsi fournies et qui, à cet effet, auront été emmagasinées dans une mémoire centrale de type RAM.

Pour correspondre avec le monde extérieur, c'est-à-dire essentiellement avec l'opérateur qui lui fournira des instructions et des données, mais aussi à qui il aura à rendre compte de ses actions, il faut ajouter un élément : l'interface d'entrée-sortie. Cet élément assure le couplage de notre système avec les différents organes de dialogue, dits périphériques, dont nous aurons besoin pour entrer en relation avec lui.

Claviers, écrans vidéo, magnétophones à cassette, imprimantes constituent l'essentiel de ces périphériques.

Une fois ce micro-ordinateur défini dans ses grandes lignes, il subsiste la question de son mode de programmation. Ce point doit être considéré en association avec l'usage auquel on le destine. Selon que l'on s'oriente vers une application domestique, industrielle ou commerciale, la réponse peut s'en trouver modifiée.

Pour conclure ce séminaire, il importe de présenter aux futurs utilisateurs de micro-ordinateurs les différents types existants sur le marché afin qu'ils puissent acquérir des éléments de choix en fonction des applications auxquelles ils les destinent et dans les limites de leur budget. Ceci nous amenant tout naturellement au débat de clôture de cette première journée « Micro-ordinateur ».



Constitution générale d'un micro-ordinateur.

Ainsi, **EN FIN D'APRÈS-MIDI** (de 16 h à 18 h) prendra place un débat sur le choix d'un micro-ordinateur à usage personnel.

Au préalable, constructeurs, importateurs et distributeurs de systèmes viendront vous entretenir des caractéristiques et des possibilités de leur produit.

Durant cette présentation, l'auditeur ne devra pas être surpris par la grande diversité qui existe déjà pour ce type de matériel.

Si la construction de micro-ordinateurs sous forme de cartes ou de boîtiers compacts, par exemple, constitue l'élément de différenciation le plus visible, on ne peut se limiter à cette seule caractéristique pour appréhender les possibilités de ceux-ci.

Ce qui veut dire que, dans son choix, le futur acquéreur aura à apprécier, en connaissance de cause, les performances du système qui tiendront la place la plus importante au moment de sa décision finale.

Bien sûr il y a la question du prix. Mais qu'on ne s'y trompe pas, ceci n'est qu'une étape.

Rapidement, nous arriverons à une concurrence telle que ce facteur sera fatalement uniformisé. Par contre, les différences continueront d'apparaître dans les performances, mais à celles-ci il faudra ajouter toutes les options qu'il sera possible de raccorder au système de base.

A ce stade, le facteur coût sera devenu secondaire. Les constructeurs s'affronteront par le biais d'options et de services offerts avec le produit.

Le hasard a voulu qu'en termes économiques, ces éléments promotionnels soient appelés **« qualités périphériques »** du produit.

Il apparaît clairement, à la lumière de ceci, que devant un tel éventail proposé à l'utilisateur, celui-ci devra, en plus d'une bonne formation sur ce sujet, disposer d'une large information afin qu'il puisse identifier le matériel correspondant à son problème.

Tout ce qui précède justifie pleinement l'ouverture d'une « journée exceptionnelle micro-ordinateur ».

Pour appuyer tous ces propos, cet ensemble de conférences conduira à un débat pour lequel trois modèles de micro-ordinateurs ont été retenus : il s'agit du TRS.80, de l'APPLE II et du PET.

Alors que chacun de ces systèmes ne manquera pas de rassembler tantôt des éloges, tantôt de sévères critiques, le but de cette tribune libre n'est pas d'élire celui qui, de ces trois appareils, pourrait recueillir un maximum de suffrages. Une telle conclusion ne correspondrait à rien d'objectif et ne serait pas réaliste.

En fait, chaque utilisateur doit partir d'un besoin qu'il lui faut analyser.

En fonction de celui-ci, il pourra déterminer les caractéristiques et performances que doit revêtir son futur micro-ordinateur. Et, comme l'élément « prix » est encore à ce jour un point suffisamment sensible, il ne lui restera plus qu'à fixer son budget, ce qui lui permettra en final de n'avoir à choisir que parmi un nombre très limité de modèles.

Si, au moment de l'ultime décision, notre objectivité est momentanément troublée par la finition, l'esthétique d'un de ces appareils, nous savons que nous pouvons nous égarer en toute sécurité à l'intérieur des choix, offrant ainsi au vendeur le sentiment qu'il a su nous convaincre et conservant la merveilleuse sensation d'avoir acheté un appareil qui nous plaît.

Jean-José WANÈGUE

Six leçons pour programmer

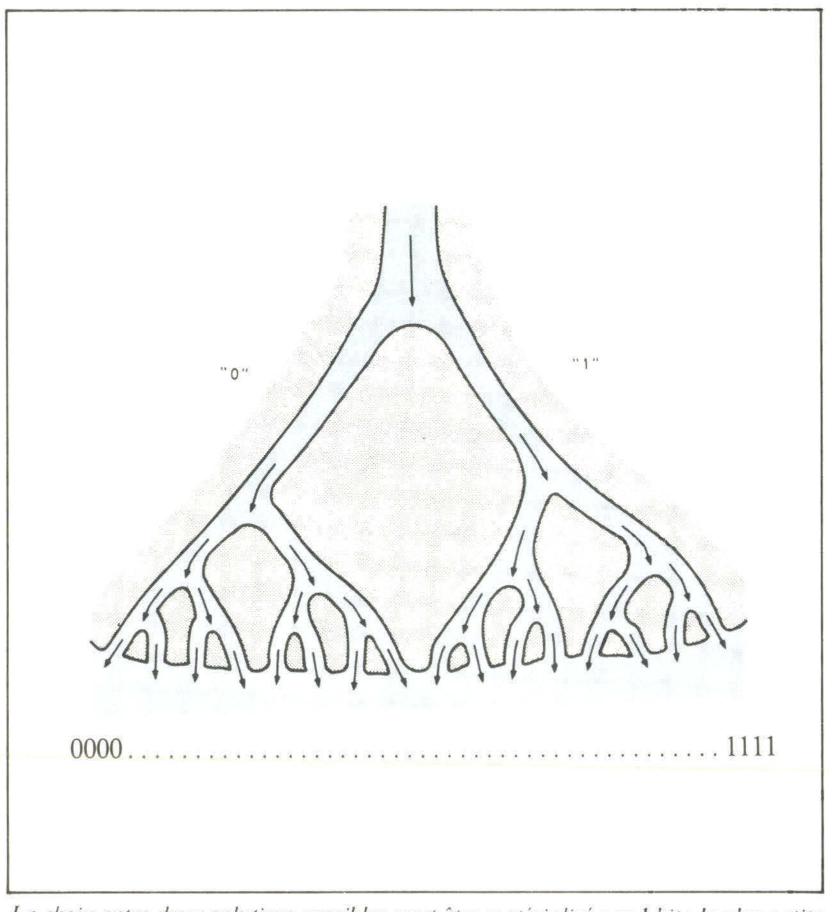
Il est convenu de dire qu'au niveau le plus élémentaire, les ordinateurs « comptent » avec deux chiffres seulement : « 0 » et « 1 ». C'est vrai, bien sûr. Cependant, avant de donner de la notation binaire une quelconque interprétation arithmétique, il est souhaitable de faire un petit tour du côté de la notion, plus fondamentale, d'information.

Pour ce faire, à notre habitude, nous recourrons bien davantage à l'intuition, ou, si l'on préfère, au « bon sens », qu'à la théorie ; ce qui revient à commencer... par le commencement.

Dans cette deuxième leçon sur la micro-informatique, notre auteur, Jean-Michel Cour, introduit de façon simple et c'est là son talent, les notions de code binaire, octal ou hexadécimal de l'information.

Il vous invite ensuite à le suivre dans la mise en service réelle du micro-ordinateur Instructeur 50 en traitant avec vous des petits programmes et en explicitant longuement la signification des touches et interrupteurs que vous devez manipuler.

Pour conclure, l'auteur vous propose quatre exercices d'application basés sur l'étude et la recherche de nouvelles instructions puis il pose aux professionnels de l'informatique la question suivante : donnez votre définition du « temps réel ».



Le choix entre deux solutions possibles peut être matérialisé par 1 bit : la plus petite unité d'information. Lorsque nous avons la possibilité de prendre 4 options différentes comme dans ce « delta » à 4 niveaux, les coordonnées du point d'arrivée seront codés sur 4 bits.

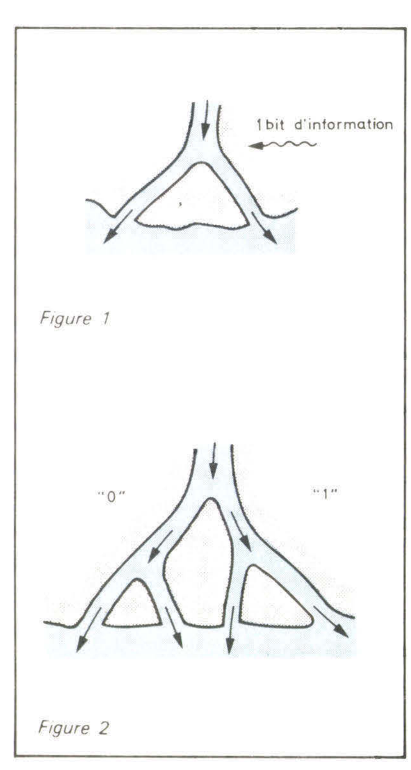
Le codage binaire de l'information

Imaginons que nous arrivons en bateau au débouché en « delta » d'une rivière **(fig. 1)**. Pour nous **aiguiller** sur l'une ou l'autre des deux branches, il faudra nous **informer** par quelque chose comme « gauche » ou « droite », ou bien « babord/tribord », ou encore (pourquoi pas si cela est **convenu** entre nous) « 0/1 ». De telle sorte que l'on nous permettra d'effectuer un **choix ;** lequel choix est bien entendu de l'espèce **la plus élémentaire,** puisque nous n'avons que **deux** possibilités. De la même façon, un référendum est un acte démocratique strictement minimum, avec « OUI/NON » ; et les jeux radiophoniques les plus simples procèdent par « VRAI/FAUX », etc.

Pour déterminer un **choix binaire** (autre manière de nommer un choix minimum), on dit en manière de définition, qu'il faut apporter un **bit** d'information ; ce bit représente, cela va de soi, **la plus petite unité d'information** pratique. Pour « matérialiser » un bit d'information, tout dispositif physique à deux états peut bien sûr faire l'affaire ; citons : un commutateur (ouvert/fermé), une lampe (allumée/éteinte) ou, comme cela se pratique dans les circuits « logiques » les plus usuels, une tension (niveau « haut »/niveau « bas »).

Allons un peu plus loin avec un « delta » un peu plus complexe **(fig.2).** Celui-ci a **quatre** embouchures, et il est clair que, pour nous aiguiller, on devra nous indiquer **deux** choix successifs, puisque chaque parcours possible comporte deux embranchements.

On peut rendre la même figure un peu plus schématique, en la dessinant autrement **(fig. 3).** On choisit arbitrairement de désigner par



Un octet, ou nombre binaire de 8 bits, permet de représenter des grandeurs numériques entre – 128 et + 127 à mieux que 1 % près.

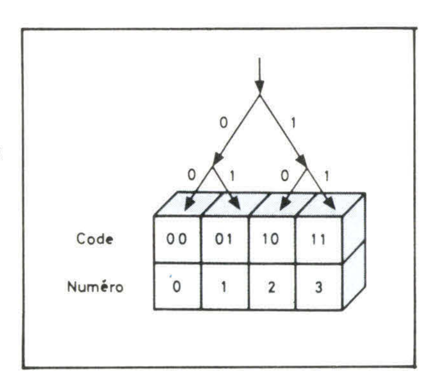


Fig. 3. – Le choix entre deux fois deux directions est codé sur deux bits. A chaque code binaire correspond le numéro décimal de la voie correspondante.

« 0 », le « tribord » du navigateur, et par « 1 », « babord ». Le dessin sera dorénavant appelé un **arbre,** en accord avec les usages en informatique.

Avec cette convention, on voit bien que le chemin « 0 puis 0 » nous amène à déboucher tout à fait à gauche ; que le chemin « 0 puis 1 » nous porte au débouché suivant, etc. L'opération, très simple dans son principe, qui consiste à « noter » le parcours effectué par une suite de choix successifs, s'appelle en bonne théorie un **codage.** Ainsi, le « chemin numéro deux » sera **codé** « 10 ».

Et l'on voit bien, intuitivement, pourquoi l'opération « inverse » qui consiste, muni du code « 11 » à trouver le « point de chute numéro trois » s'appelle un **décodage.**

Ceci dit, notre dessin n'était pas si arbitraire que cela ; en particulier, nous nous sommes « arrangés » pour que le chemin désigné par des « 0 » nous amène au « débouché zéro ». Ce, pour introduire le **« codage naturel »** des nombres entiers.

A ce niveau, le lecteur qui a déjà un peu fréquenté les notations binaires pourra passer à la section suivante ; sinon, **nous insistons** pour qu'il prenne quelques minutes pour annoter, à la manière de notre premier arbre, les arbres à trois et quatre niveaux des **figures 4 et 5.**

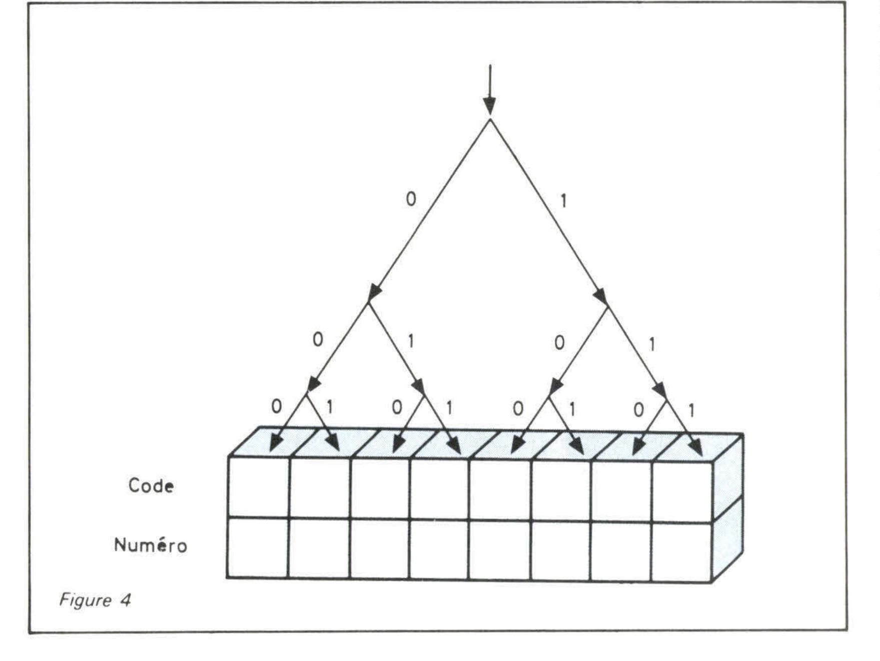


Figure 4

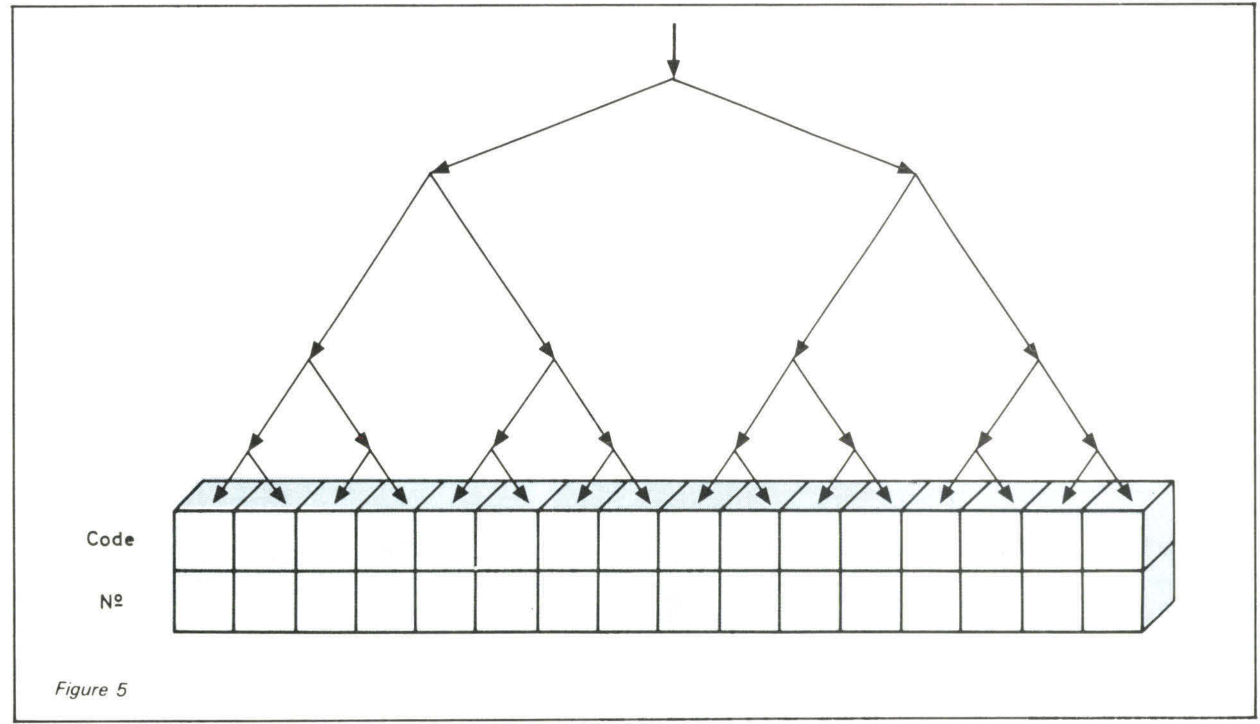


Figure 5

Fig. 4. et 5. – Inscrivez respectivement, à titre d'exercice, dans chacun de ces « arbres » à 3 et 4 niveaux le code binaire et décimal correspondant à la direction suivie.

Relation avec l'addition binaire

Introduisons maintenant un peu d'arithmétique ; si nous ne disposons **que** de deux chiffres, le « 0 » et le « 1 », on imagine sans difficulté que la table d'addition convenable est celle de la **figure 6.** Ce que l'on vérifie sur la même figure, en faisant l'addition de « un », successivement, à « zéro », « un », « deux », et en constatant (stupeur !) que l'un retrouve les codes binaires successifs des quatre premiers nombres entiers. Autrement dit, cette table permet de **compter** correctement.

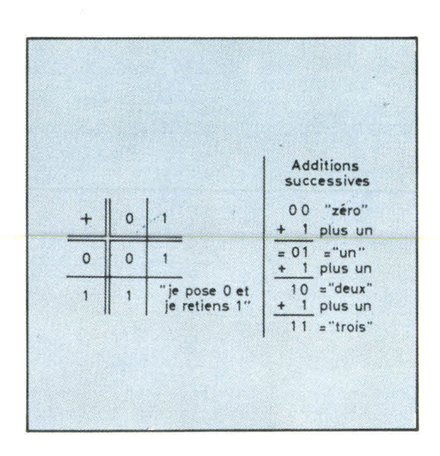


Fig. 6. – Table d'addition binaire.

Ce constat est moins bête qu'il n'en a l'air. En effet, **tous** les ordinateurs fonctionnent grâce à ce jeu de relations simples entre l'arithmétique et les mécanismes de codage/décodage de l'information. En particulier, les instructions successives d'un programme sont désignées par un **compteur** qui donne des numéros successifs (des **adresses** au sens introduit dans le précédent article) ; ces adresses sont ensuite **décodées** grâce à des éléments qui « matérialisent » très précisément un « arbre » comme ci-dessus, donnant accès à une position de mémoire déterminée.

Quelques notions importantes

A l'aide de **N** bits, **combien** d'« objets » différents savons-nous **compter,** c'est-à-dire, car nous savons maintenant que c'est la même chose, combien d'objets savons-nous **représenter ?**

A l'aide d'arbres du genre que nous avons employés, il n'est pas difficile de voir que **chaque bit** (autrement dit, chaque choix) **multiplie par deux** nos possibilités. D'où se déduit la « loi ».

« AVEC **N** BITS, ON PEUT CODER $\mathbf{2^N}$ OBJETS ».

Ce qui vous explique, entre autres, pourquoi nous avons dit précédemment qu'avec 15 lignes d'adresse (chacune matérialise un bit) notre microprocesseur 2650 pouvait désigner 32768 positions de mémoire.

Nous recommandons au lecteur, à ce stade, de consulter attentivement le grand tableau de la **figure 7,** dans lequel ont été reportés :

— les « nombres de bits » les plus en usage, c'est-à-dire que l'on rencontre « groupés » dans la pratique,

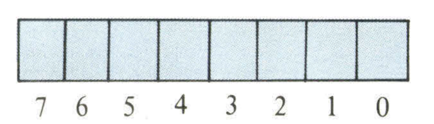
— le nombre de possibilités qui leur correspond, et l'usage qui en est fait, classiquement, en informatique (micro- ou non).

Notre unité de travail : l'octet

On constate, sans surprise pour celui qui est familier de l'historique des ordinateurs, que l'**octet *,** qui est, rappelons-le, le « paquet » de huit bits, s'est largement imposé comme unité pratique de travail, grâce à sa très grande versatilité. Ceci est vrai, tout particulièrement, pour les **microprocesseurs « tous usages »,** famille à laquelle notre 2650 appartient.

Le tableau **(fig. 7)** nous donne déjà un grand nombre d'exemples où il s'emploie. Il permet en outre par une « astuce » de représenter des nombres avec signe, entre – 128 et + 127, autrement dit, des grandeurs « numériques » à mieux que 1 % près.

Pour désigner les bits d'un octet, la meilleure habitude est de les numéroter de droite à gauche, avec les chiffres zéro à sept (*). On représente souvent l'octet comme suit :



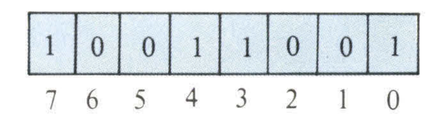
N	2^N	Sortes d'objets que l'on peut représenter
1	2	Contact (ouvert/fermé), lampe (éteinte/allumée)
4	16	Les chiffres décimaux 0, 1, ..., 9 (il reste six possibilités inemployées)
6 ou 7	64 128	« Signes » d'un « alphabet » au sens large : « lettres », « chiffres », « ponctuation »...
8	256	— Nombres de 0 à 255 — « gros » alphabet — Paire de chiffres décimaux, en considérant deux fois 4 bits (et en laissant de côté 156 possibilités)
10	1024	« Presque mille » ; les professionnels notent K cette quantité ; certains lisent « kilo »
16	65536	D'assez grands nombres entiers

Fig. 7

* *L'octet est le plus souvent nommé « byte » dans la littérature de langue anglaise. Mais aussi « 8 bits word »*

* *Curieusement, la société IBM a adopté la numération de gauche à droite, ce qui ne va pas sans poser des problèmes quelquefois...*

où chaque « case » contient un bit. Cette convention est la plus naturelle, car elle est en relation directe avec le codage binaire des nombres, comme cela apparaît sur l'exemple suivant :



Si nous avions le courage (et la place !...) de dessiner un arbre avec huit niveaux, nous verrions à l'évidence que cet octet représente « cent cinquante trois » ; nous pouvons aussi remarquer que

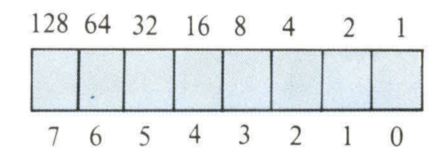
$$153_{(décimal)} = (\mathbf{1} \times 2^7) + (\mathbf{0} \times 2^6) + (\mathbf{0} \times 2^5) + (\mathbf{1} \times 2^4) + (\mathbf{1} \times 2^3) + (\mathbf{0} \times 2^2) + (\mathbf{0} \times 2^1) + (\mathbf{1} \times 2^0)$$

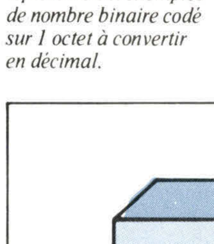
soit, en enlevant les termes nuls,

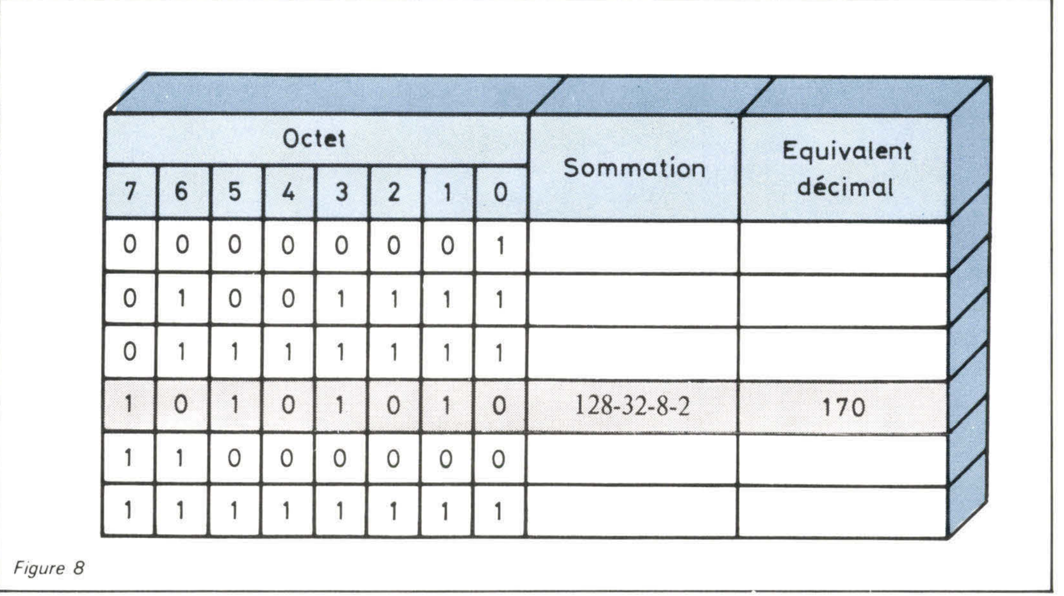
$$153 = 128 + 16 + 8 + 1$$

Cette décomposition est très générale, et se justifie avec d'excellentes mathématiques... nous retiendrons simplement que chaque position a un **« poids »** binaire : le bit le plus à droite « pèse » un (2 puissance 0) ; le bit suivant « pèse » deux (2 puissance 1) ; etc., jusqu'au bit le plus à gauche qui « pèse » cent-vingt huit (2 puissance 7).

Ainsi, pour convertir un octet en son équivalent décimal naturel, suffit-il de retenir, en regard de chaque position, son « poids ».




Fig. 8. - Ce tableau représente six exemples de nombre binaire codé sur 1 octet à convertir en décimal.



Octet								Sommation	Equivalent décimal
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1		
0	1	0	0	1	1	1	1		
0	1	1	1	1	1	1	1		
1	0	1	0	1	0	1	0	128-32-8-2	170
1	1	0	0	0	0	0	0		
1	1	1	1	1	1	1	1		

Figure 8

Fig. 9. - Conversion du nombre décimal 162 en binaire par soustraction successive.

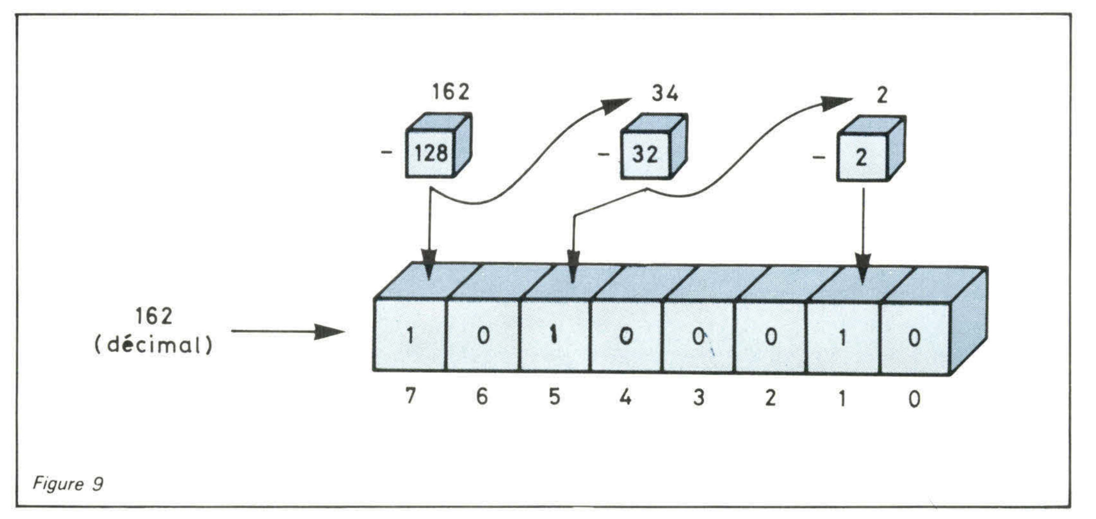


Figure 9

Si l'on souhaite s'exercer, on peut garnir un tableau comme celui de la **figure 8,** dans lequel une ligne a déjà été remplie à titre d'exemple.

Pour les nombres décimaux de 0 à 255, la conversion inverse s'effectue aisément, par soustractions successives de « la plus grande puissance de deux inférieure », comme dans l'exemple de la **figure 9.**

La notation hexadécimale

L'esprit humain est ainsi fait qu'il énonce, et qu'il retient plus aisément des notations condensées (comme, par exemple, le prénom ALBERT exprimé dans un alphabet de 26 lettres), que de longues séries de zéros et de uns. De plus, nous sommes habitués à l'emploi d'une dizaine de symboles pour les nombres... pour cette simple (et seule !) raison que la nature nous a dotés de **dix** doigts.

On s'aperçoit facilement que les deux paquets de bits qui se « rapprochent » le plus de nos habitudes sont :

- le paquet de **trois** bits (**huit** valeurs),
- le paquet de **quatre** bits (**seize** valeurs).

Ne retenant que ce dernier, on associera à **chaque combinaison** de quatre bits, **un symbole ;** pour les dix premiers, dans l'ordre numérique naturel, il est bien évidemment avantageux d'employer

les chiffres décimaux habituels :

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Pour les six combinaisons restantes, on a imaginé de prendre les six premières lettres de l'alphabet

1010	A	(dix)
1011	B	(onze)
1100	C	(douze)
1101	D	(treize)
1110	E	(quatorze)
1111	F	(quinze)

Avec cette convention, les suites de zéros et de uns peuvent s'exprimer à raison d'**un** symbole pour **quatre** bits, en découpant la chaîne par tranches de quatre, **à partir de la droite** (obligation liée à la notion des poids croissants).

Ainsi

douze bits donnent		
1 1 0 1	0 0 1 0	1 1 1 1
D	2	F
trois symboles		

On appelle les signes 0, ..., 9, A, ..., F : **chiffres hexadécimaux.** Ils s'utilisent le plus souvent en lieu et place de la notation binaire pure. On peut en outre retenir que dans les textes de langue anglaise, la tranche de quatre bits est fréquemment nommée **« nibble »,** par extension d'un verbe anglais qui veut dire « grignoter », « prendre un petit morceau ».

Le meilleur moyen de nous familiariser avec toutes ces notions et notations — qu'il fallait bien introduire au préalable — est de les faire « fonctionner » sur notre micro-ordinateur I50 ; ce que nous entreprendrons sans plus attendre.

Mise en service du micro-ordinateur

Le micro-ordinateur I50 se met en service comme une simple machine à calculer : il suffit de le brancher au secteur par l'intermédiaire de son petit transformateur d'adaptation. Dès cet instant, il affiche « HELLO », ce qui signifie qu'un programme pré-enregistré, le **Moniteur,** est prêt à dialoguer avec l'utilisateur.

Le clavier principal du micro-ordinateur comporte, rappelons-le, deux ensembles de touches groupées ; à gauche, les **touches de fonction,** et à droite un clavier pour l'entrée de chiffres hexadécimaux, gravé en conséquence.

Parmi les fonctions, on repèrera tout de suite la touche MON, qui permet de « reprendre contact » avec ledit programme Moniteur. Lorsque l'on veut ; par exemple, appuyer sur une touche hexadécimale quand « HELLO » est affiché : il apparaît « Error 2 » (c'est une fausse manœuvre). On revient au Moniteur par MON, et « HELLO » réapparaît.

Adressage et lecture de la mémoire

La **mémoire vive** (RAM, pour la technique !) qui servira indifféremment au stockage des « programmes » et des « données » (la distinction, vous le verrez, n'est pas toujours évidente) est organisée à la façon d'un vaste **tableau** binaire, à « huit colonnes ». Chaque « ligne », aussi dénommée **case** ou **mot** ou simplement **position** de mémoire, contient exactement un octet d'information.

Le numéro de ligne, ou **adresse,** repère un octet que l'on peut **lire** ou **écrire** individuellement. Théoriquement, le microprocesseur 2650 permet de désigner 32 K (32 fois 1024) positions de mémoire. Dans l'I50 nous avons **512 positions** (2 puissance 9), pour notre usage.

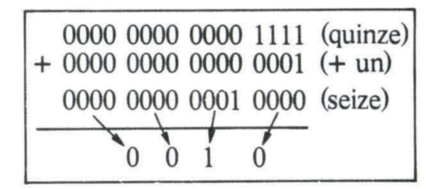
Le Moniteur nous donne le moyen de « voir » la mémoire en question, sur l'affichage, et ceci octet par octet. Pour ce faire, on presse la touche MEM, et il s'affiche « . Ad . = » ; ce qui signifie que le Moniteur attend qu'on lui donne l'adresse de la position de mémoire demandée.

On commencera par l'adresse zéro, en frappant 0 puis ENT/NEXT ; on voit alors s'afficher « 0000xx », ce qui se lit de la manière suivante :

- « 0000 » est l'**adresse** que nous avons demandée, et qui est complétée par des zéros à gauche ;
- « xx » est le **contenu** de la position de mémoire correspondante, contenu qui, à ce stade, est... n'importe quoi, parce que ce type de mémoire prend, à la mise sous tension, des valeurs tout à fait imprévisibles.

Effectuons à présent une suite de frappes sur la touche ENT/NEXT, et observons les quatre chiffres à gauche de l'affichage : on voit tout de suite qu'ils **progressent en compteur** (0, 1, 2, 3,...). Attention ! que se passe-t-il après « 0009 » si l'on poursuit ?

On obtient 000A, en hexadécimal, soit dix ; ceci correspond bien au fait que le compteur est **binaire,** et que nous en voyons l'image « condensée » hexadécimale. Poursuivant jusqu'à 000F (quinze), il vient ensuite 0010 qu'**il faut lire « seize ».** On vérifie facilement que c'est le résultat de l'addition binaire



Pour simplifier il est utile de retenir que tout se passe comme dans l'arithmétique que nous avons apprise à l'école, l'arithmétique décimale ; **à ceci près** que le report s'effectue à « seize » et non à « dix » ; l'analogie des deux exemples suivants est flagrante :

(en décimal)	(en hexadécimal)
1 9	1 F
+ 1	+ 1
2 0	2 0

Le nombre hexadécimal 00AA s'énonce « ha-ha » et représente cent soixante dix, c'est-à-dire dix fois seize plus dix.

Prenez garde ! Si vous voyez affiché 0020, vous serez très tenté de lire « vingt » ! Grave erreur, c'est « trente - deux » ! Dans un cas pareil, **il faut** s'habituer à énoncer soit « trente-deux », soit « deux-zéro », et se faire une règle d'éviter toute confusion entre les systèmes arithmétiques de la machine et ceux de la vie de tous les jours... Evidemment avec quelque chose comme 00AA on n'aura pas le problème, on énoncera naturellement « ha-ha » qui est cent soixante-dix ; ce qui se vérifie en l'exprimant comme « dix fois **seize** plus dix » de même qu'en notation décimale 83 s'interprète comme « huit fois **dix** plus trois ».

Rassurez-vous (si vous êtes inquiet), on s'y habitue très vite !

Au passage, vous aurez remarqué que les lettres B et D sont dessinées en « minuscules » (b. et d.), car le système d'affichage à sept segments (ils sont tous allumés pour 8)(8) oblige à styliser quelque peu les signes de l'alphabet. Afin d'éviter toute erreur de lecture, comme par exemple une confusion entre 6 (6) et B (b.), un **point** apparaît avec les lettres « dangereuses ».

Ecrire en mémoire

Revenons à l'adresse zéro par

MON MEM 0 ENT/NEXT

et frappons (lentement) la suite 0, 1, 2, 3, 4,... On voit alors que les deux positions les plus à droite sur l'affichage contiennent successivement : 0, 01, 12, 23, etc.

Cela veut dire que le Moniteur prend en compte les frappes un peu à la manière d'une calculatrice de poche ; les chiffres « entrent par la droite », et « disparaissent à gauche ». Deux chiffres hexadécimaux seulement demeurent, car nous sommes censés donner **exactement un octet** d'information. A la première frappe, un seul chiffre apparaît ; dans ce cas, le Moniteur considère qu'il est précédé d'un « zéro fantôme ».

Pour refaire les choses dans l'ordre, revenons à l'adresse zéro par la première séquence

MON MEM 0 ENT/NEXT

puis frappons

7 0 ENT/NEXT

L'afficheur donne maintenant l'adresse 0001, ce qui signifie que :

- nous avons **écrit** 70 à l'adresse 0000,
- que nous « voyons » maintenant l'octet d'adresse 0001,
- que le Moniteur attend une nouvelle manœuvre.

Revenons à l'adresse zéro une fois de plus, toujours par le même procédé. On voit alors 0001 70 ; la mémoire a « retenu » notre octet de valeur 70. Passant à l'adresse 0001 par ENT/NEXT, nous écrirons la suite d'octets suivants :

F 0 ENT/NEXT

4 0 ENT/NEXT

Nous pouvons « relire » (et par là même, vérifier) cette séquence, en revenant à l'adresse zéro, puis en « défilant » grâce à ENT/NEXT. Il vient

0000	70
0001	F0
0002	40
0003	xx

(nous n'avons encore rien mis en 0003).

Stop ! Revenons au Moniteur par MON et ne touchons plus à rien ! Nous venons d'introduire dans l'I50 notre premier programme...

La définition d'un programme

Nous avons déjà dit dans notre première partie, qu'un ordinateur exécute des **suites** d'ordres élémentaires, ou **instructions,** lesquelles suites d'instructions sont enregistrées dans une mémoire. Il faut pour ce faire que l'ordinateur dispose d'un **compteur** qui progresse automatiquement en séquence, à la manière de celui qu'affiche le Moniteur pour nous permettre de « lire » ou d'« écrire » des positions de mémoire d'**adresses successives.**

Cet organe, que l'on appelle en bon français **compteur ordinal** et en anglais « program counter » (le plus souvent abrégé en « PC »), fait partie intégrante de notre microprocesseur 2650. Par définition même, le PC progressera de telle sorte qu'il contienne l'**adresse de l'instruction suivante.**

Chaque étape ou **pas de programme** (program **step,** en anglais) consiste grossièrement en ceci :

- le microprocesseur **lit, à l'adresse donnée par le PC,** un octet qui constitue le **code** d'une **opération** parmi celles prévues par le fabricant (entrée, sortie, calcul, recopie...) ;
- à l'intérieur du microprocesseur, l'instruction est... **décodée** pour actionner les éléments convenables (dispositif d'entrée, additionneur, mémoire...) ;
- quasi **simultanément,** le microprocesseur fait progresser le PC de un.

Pour la programmation, seul compte le **résumé** de l'opération, autrement dit l'effet « global » d'une instruction sur l'« état des lieux » ; on pourrait dire encore : la **relation** entre l'état du micro-ordinateur **avant** et **après** l'instruction. Cette relation est entièrement déterminée par le **code-opération** choisi.

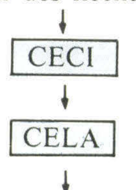
D'autre part, le programmeur retient simplement que, sauf exception **voulue,** le microprocesseur « enchaînera » automatiquement sur l'instruction qui suit (en mémoire).

Ces enchaînements peuvent s'exprimer de plusieurs façons :

(1) par du **texte** FAIRE CECI
FAIRE CELA

(2) par un dessin conventionnel dit **organigramme** dans lequel les actions s'expriment dans des

rectangles, et leur enchaînement par des flèches



(3) par la suite des codes-opération, codés en **binaire,** ou, plus commodément, en hexadécimal (ce qui revient au même), ces codes étant disposés en regard de l'adresse où ils sont implantés dans la mémoire.

Adresse	Code-op
0000	70
0001	F0
0002	40

Nous pouvons décrire le tout petit programme que nous venons d'entrer, de ces différentes manières :

1 - Texte

- « Entrer l'octet constitué par les commutateurs du pupitre »
- « Sortir cet octet sur les voyants »
- « Arrêter l'ordinateur ».

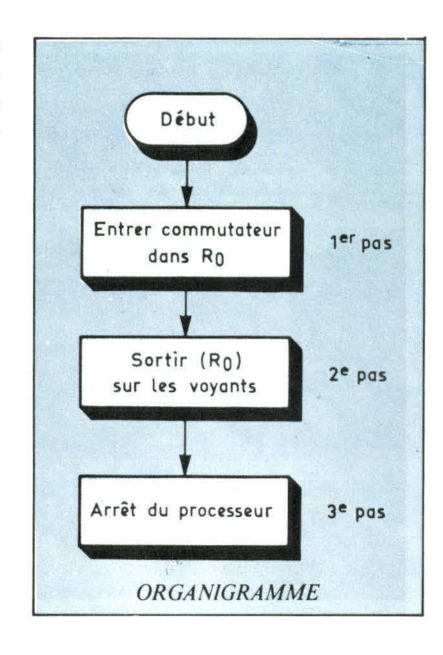
Fig. 10. - Circulation des informations entre les commutateurs et les voyants via le registre interne Ro du microprocesseur.

Les actions d'entrée et de sortie nécessitent l'intervention d'un élément **pour mémoriser temporairement** l'octet, entre les deux premiers pas. Le microprocesseur dispose pour ce faire d'une petite mémoire incorporée distincte de la « mémoire principale » (celle que nous avons manipulée précédemment). Ses cases sont baptisées **registres** pour bien les distinguer. Une opération « entrée » consiste à recopier un octet venu du monde « extérieur » (ici, les commutateurs) dans un de ces registres. Une opération de sortie, consiste à recopier le contenu d'un registre dans un organe « extérieur » (ici : les voyants).

Nous utiliserons ici le registre numéro zéro (abrégé en RO) comme intermédiaire ; la circulation de l'information peut se résumer par le petit diagramme de la **figure 10.**

2 - Organigramme

On utilise la notation : (RO) qui se lit « contenu du registre zéro » afin de distinguer le contenu (information) du contenant (registre) ; il ne faut pas confondre le liquide et la bouteille !



ORGANIGRAMME

3 - Programme binaire

A condition de connaître le montage de l'I50 et le **répertoire** (le « menu » si vous préférez) des instructions du microprocesseur 2650, on sait que :

— l'instruction du code 70 recopie les huit commutateurs dans le registre zéro, à condition que le commutateur à trois positions placé dessous soit dans la position « NON EXT DATA PORT » (faisons le tout de suite avant de n'y plus penser) ;

— l'instruction de code F0 recopie

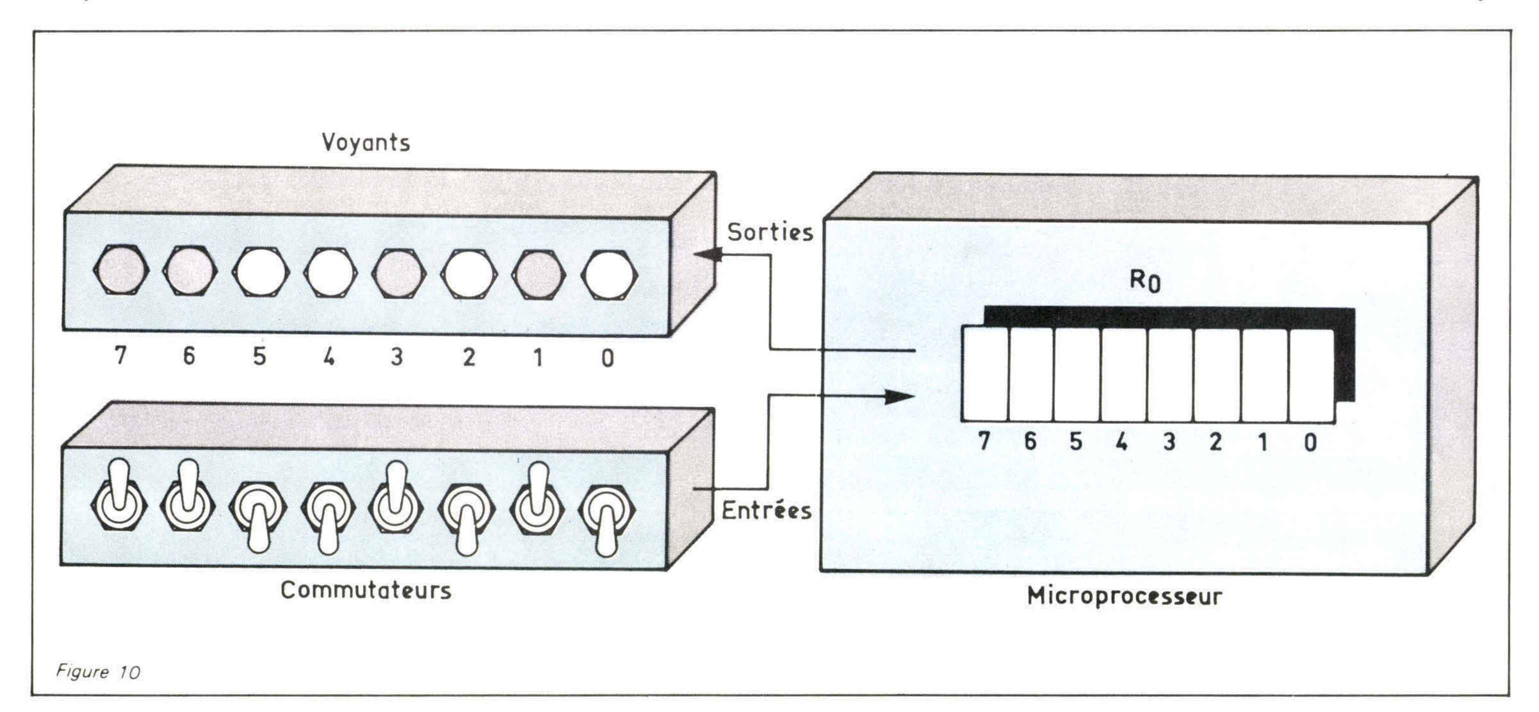


Figure 10

Le bouclage d'un programme est effectué par un ordre qui force le compteur ordinal à une valeur qui n'est pas l'adresse suivante.

le contenu du registre zéro sur les voyants (à la même condition) ;
— l'instruction du code 40 **halte** (bloque) le microprocesseur.

On peut alors écrire le programme binaire en le documentant par ses adresses d'implantation (à gauche) et un « commentaire » (à droite).

Adresse	Code op.	Commentaire
0000	70	RO ← commutateurs
0001	F0	Voyants ← RO
0002	40	Halte

On remarquera la notation

« destination » ← « source »

qui est extrêmement commode (et rigoureuse) pour dénoter les opérations de **recopie** d'information, plus généralement baptisées **affectations ;** la seconde instruction « affecte aux voyants le contenu du registre zéro ».

L'exécution d'un programme

La manière la plus rapide de faire marcher notre programme est de frapper la touche RST (abréviation de l'anglais reset, qui veut dire : restauration). Avant de le faire, mettez les commutateurs dans une position quelconque, frappez RST et vous observerez deux choses :

— les commutateurs sont « recopiés » sur les voyants en regard,
— le voyant séparé « RUN » s'est éteint, ce qui traduit **l'arrêt** du microprocesseur.

Pour reprendre le contact avec le moniteur, il vous faudra dans ce cas enfoncer **simultanément** les touches RST et MON.

On peut recommencer l'opération un certain nombre de fois pour plusieurs positions des commutateurs ; il est inutile de revenir au moniteur, une frappe sur RST permet l'exécution à chaque fois.

L'effet de RST s'explique comme suit : cette frappe provoque la **restauration** du microprocesseur 2650 et en particulier la remise à zéro du PC (compteur ordinal) ; actionné par les impulsions de son horloge, le 2650 exécute aussitôt l'instruction à cette adresse 0000 (lecture des commutateurs) puis celle de l'adresse 0001 (écriture sur les voyants) et enfin, en 0002 celle qui l'arrête.

Comme chaque instruction s'exécute en quelques microsecondes, le programme nous paraît visuellement instantané. Il nous est possible, si nous le voulons, de voir cette exécution « au ralenti » ou plus précisément « pas à pas » à l'aide du moniteur.

L'exécution pas à pas

Pour ce faire, revenons au moniteur par RST MON simultanées et frappons cette fois la touche REG. Il apparaît maintenant le message « r = » qui est une invitation à consulter ou modifier le **registre** interne du microprocesseur et non plus la mémoire principale. La « logique » de cette consultation/modification est sensiblement la même que pour la mémoire.

Si l'on frappe 0, il doit apparaître « r0 = xx » (où xx est une valeur qui dépend de ce que nous venons de faire). Cette manœuvre est celle qui permet de **consulter** le contenu d'un registre. Si on le modifie par une frappe hexadécimale comme C F ENT/EXT on voit apparaître « r1 = xx », ce qui donne accès au registre 1, etc. On peut alors relire le registre 0 par la manœuvre REG 0 et l'on retrouve « r0 = CF ».

Le compteur ordinal est lui aussi un « registre » interne du 2650 et comme tel on y accède par REG C (C comme Compteur). On observe alors l'affichage « PC = xxxx » (PC comme **Pro**gram **C**ounter) avec une valeur à **quatre chiffres ;** c'est que ce compteur est de plus grande capacité que les autres registres puisqu'il doit pouvoir « compter jusqu'à 32 K » et comporte **quinze positions** binaires ; d'où il s'ensuit qu'il faut quatre chiffres hexadécimaux pour donner sa valeur.

Pour exécuter le programme en pas à pas à partir de l'adresse zéro, il nous faut **l'initialiser** par la manœuvre REG C 0 ; il apparaît alors « PC = ... 0 » qui signifie que le programme commence à s'exécuter à cette adresse. Inscrivons avant de poursuivre une configuration choisie sur vos huit commutateurs :

« 1 »
« 0 » 7 6 5 4 3 2 1 0

qui s'interprète en binaire comme 1100 1001 ; pressons maintenant la touche STEP Il apparaît maintenant affiché 0001 F0 qui s'interprète comme suit :
— 0001 est le contenu de PC (adresse de l'instruction suivante).
— F0 est le code opération situé à cette adresse, autrement dit celui qui sera exécuté **au pas suivant.**

Avant de poursuivre, nous pouvons grâce au moniteur « regarder » le registre zéro par REG 0, et nous trouvons « 0 = C9 », autrement dit nos commutateurs traduits en hexadécimal. Frappant maintenant STEP on obtient 0002 40 dont l'interprétation doit être claire d'après notre description du premier pas. Les voyants affichent maintenant la recopie des commutateurs.



Si l'on regarde à nouveau le registre zéro (même manœuvre que ci-dessus), on constate qu'il contient toujours C9 ; un registre est bel et bien une mémoire, et comme telle, n'est pas modifié quand il est la **source** d'une recopie ; seule la **destination** (par définition !) est **affectée.**

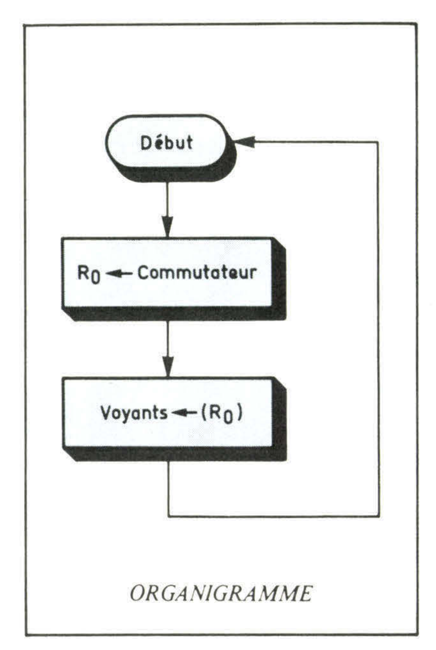
Effectuons le dernier pas par STEP ; cette fois-ci **tout** s'éteint ! C'est normal car nous avons forcé un arrêt complet du système... Moniteur compris.

Bouclage du programme

Cet arrêt du processeur est avouons-le frustrant ; il nous oblige à « manipuler » pour renouveler l'exécution du programme. Afin de rendre ce programme un peu plus intéressant, nous souhaiterions le décrire autrement.

● **(1) Texte,** début :
— entrée des commutateurs
— sortie sur les voyants
— recommencer.

● **(2) Organigramme,** on dessine une flèche qui revient au début, créant ainsi ce que l'on appelle une **boucle** dans le programme.



ORGANIGRAMME

● **(3) Programme binaire,** à l'emplacement de mémoire 0002, il va bien falloir inscrire autre chose qu'une instruction qui arrête le microprocesseur. Nous pourrions imaginer d'écrire une nouvelle suite d'instruction d'entrée puis de sortie et ainsi de suite.

0000	70	entrée
0001	F0	sortie
0002	70	entrée
0003	F0	sortie
0004	70	entrée
etc.		

Cette solution n'est cependant pas satisfaisante car si le PC progresse indéfiniment, nous allons évidemment « sortir » de la mémoire (qui n'est pas infinie, mais limitée sur l'I50 à 512 octets).

Ce qu'il nous faut c'est bien un ordre qui force le compteur ordinal à une valeur (ici : 0000) **qui n'est pas l'adresse suivante ;** ce qui se définit comme une **rupture de séquence.** Pour ce faire, les microprocesseurs possèdent des instructions particulières qu'on appelle **sauts** (anglais : **jump)** ou **branchements** (anglais : **branch).**

De telles instructions ont pour effet non pas de manipuler les données, mais d'**altérer la séquence** du programme ; et nous verrons que c'est par ce biais que l'on peut programmer des **décisions,** autrement dit, rendre, les ordinateurs un peu « intelligents ».

Pour notre propos, nous trouverons dans le répertoire du 2650 une instruction qui se codera 1F 00 00, c'est-à-dire avec **trois octets** (celles que nous avons vues n'en comportaient qu'un). Le premier est le **code-opération** proprement dit ; en l'occurrence « recopier dans le PC les deux octets suivants » ; les deux octets suivants, que l'on appelle judicieusement la **partie-adresse** de l'instruction, sont précisément la valeur de l'adresse du programme (zéro) où nous voulons revenir. D'où (enfin) notre programme codé en binaire :

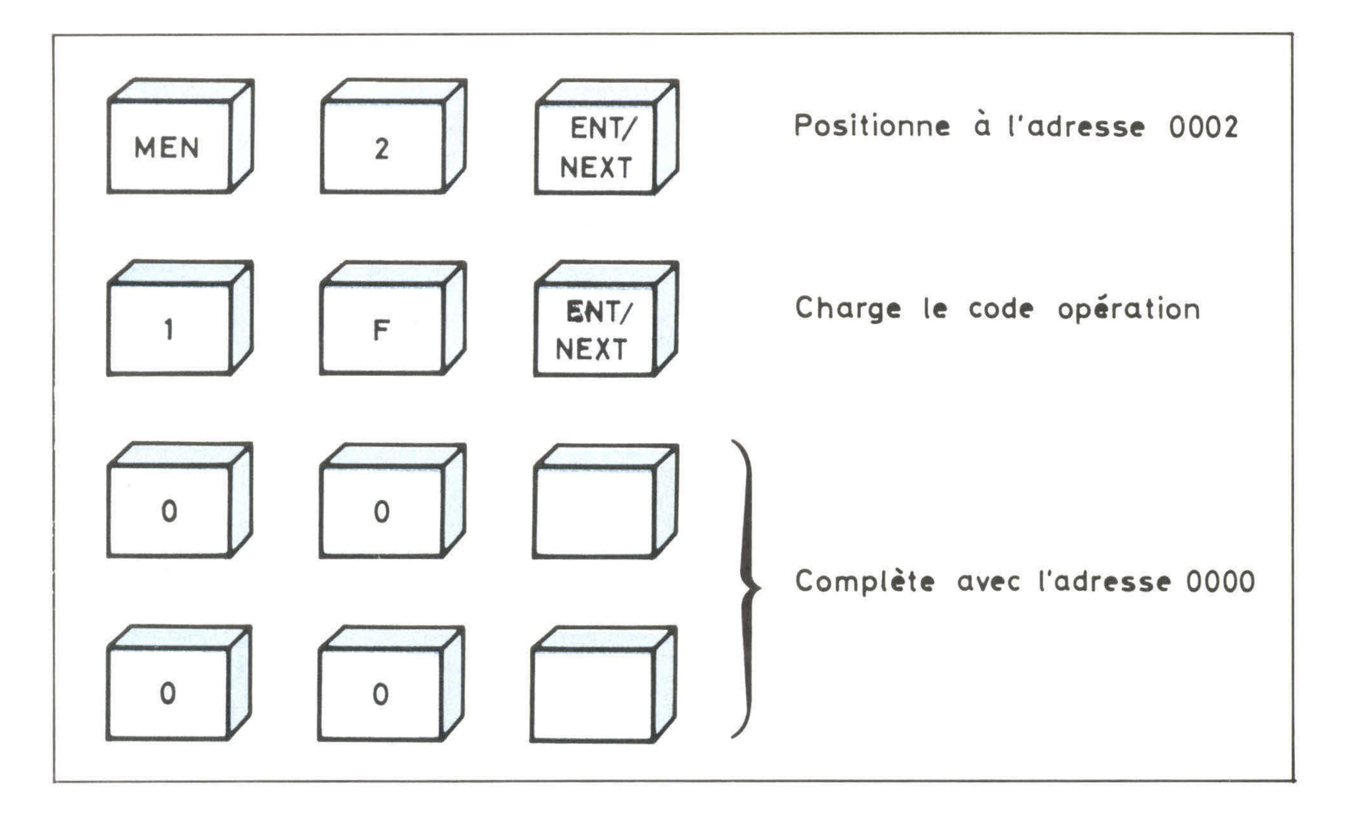
Adresse	Code	Commentaire
0000	70	RO ← commutateur
0001	F0	Voyants ← (RO)
0002	1F	Saut
0003	00 }	à l'adresse 0000
0004	00 }	

Revenons au moniteur (MON ou MON + RST) et **modifions** le programme par la **figure 11.**

L'exécution « naturelle » par RST nous fait maintenant constater que les voyants changent « instantanément » (c'est-à-dire en quelques microsecondes !) quand on change un commutateur. Le voyant RUN ne s'éteint plus. ON peut ainsi suivre comme précédemment le programme en pas à pas et constater qu'après avoir atteint le pas de l'adresse 2 :

0002	1F	(code du « saut »)

Fig. 11. – Dans cet exemple, l'instruction 1F (saut) force le compteur ordinal à l'adresse 0000.



STEP nous ramène au pas de l'adresse 0

0000 70

Dans toute la suite, nous tâcherons de nous astreindre à expliciter les relations **directes** qui existent entre ces niveaux de description, qui sont caractéristiques de la démarche **d'analyse-programmation** et complètent une véritable « technologie * » de la mise en place de logiciels.

Pour ceux qui ont un matériel et du courage, ils peuvent tout de suite pratiquer les quelques exercices très simples qui sont proposés.

Exercices

I – On insère une nouvelle instruction (de deux octets celle-ci) entre l'entrée des commutateurs et la sortie sur les voyants. **Sans regarder** les documents de référence :

Quelques points à retenir

Nous avons vu à travers nos exemples très simples :

— La forme sous laquelle l'information est représentée et manipulée dans le micro-ordinateur.

— Le mécanisme de l'exécution en séquence à l'aide d'un compteur qui désigne les instructions successives.

— L'existence d'instructions de longueur différente que l'on prend dans un « répertoire » (l'objet du prochain chapitre sera de décrire le répertoire de notre microprocesseur).

— Les mécanismes par lesquels on peut en pratique consulter et modifier différents éléments de mémoire (mémoire principale, registres) à l'aide du moniteur.

— Que nous pouvons déclencher l'exécution du programme de plusieurs façons ; et en particulier la « surveiller » en pas à pas grâce au même moniteur.

En outre, nous avons pu entrevoir trois NIVEAUX de description d'un programme.

— **TEXTE** en français (peut être pas très littéraire).

— Présentation par boîtes et fiches **(ORGANIGRAMMES).**

— Présentation finale par un programme **CODÉ BINAIRE.**

* *Afin qu'on ne nous accuse pas d'abus de vocabulaire : technologie :... étude des techniques, des outils, des matériaux (Petit Robert).*

(a) Trouver en l'exécutant quelle est l'opération arithmétique (simple) qui s'effectue :

Adresse	Code bin.	Commentaire
0000	70	RO ← commutateur
0001	84	?
0002	01	
0003	F0	Voyants ← (RO)
0004	1F	
0005	00	Retour à 0000
0006	00	

(b) Puis, changer la valeur en 0002 et interpréter les résultats.

(c) Remettre en 0002 la valeur 01 ; partant de tous les commutateurs en position zéro (vers le bas du pupitre), les lever successivement de droite (position 0) à gauche (position 7) ; observer et interpréter le résultat.

II – Pour ceux qui ne sont pas habitués aux conversions (binaires/hexa) exécuter en pas à pas le programme commutateurs/voyants du texte :

(a) Essayer de prédire « de tête » l'équivalent hexadécimal des commutateurs, et vérifier dans le registre zéro au pas 0001 F0.

(b) A l'inverse, entrer une valeur hexadécimale dans le registre zéro par modification (cf. le texte) et prédire ce qui apparaîtra sur les voyants.

III – L'instruction à un seul octet de code C0 « ne fait rien » (non operation). Entrez le programme suivant dans le micro-ordinateur, sans regarder les documents de référence.

Adresse	Code bin.	Commentaire
0000	C0	
0001	C0	
0002	C0	
0003	70	
0004	F0	
0005	1F	
0006	00	
0007	03	

(a) Vérifier que ce programme fait bien le même travail que celui du texte.

(b) A l'aide d'une exécution en pas à pas, interpréter la modification.

(c) Faites l'organigramme correspondant.

IV – On insère dans le programme des instructions (dont le détail n'est pas important pour l'exercice) et qui introduisent un délai notable (de l'ordre de la seconde) entre l'entrée des clés et l'affichage ; entrer et exécuter par RST :

Adresse	Code	Commentaire
0000	70	RO ← commentaire
0001	7A	
0002	7E	Délai
0003	7B	
0004	7C	
0005	F0	Voyants ←
0006	1F	
0007	00	Retour à 0000
0008	00	

(a) Que se passe-t-il quand on bouge rapidement les commutateurs ? Lentement ?

(b) A quelle condition les voyants sont-ils l'image **fidèle retardée** des commutateurs ?

(c) Généraliser en se souvenant que le microprocesseur met toujours « un certain temps » à passer d'une entrée à une sortie (délai des instructions).

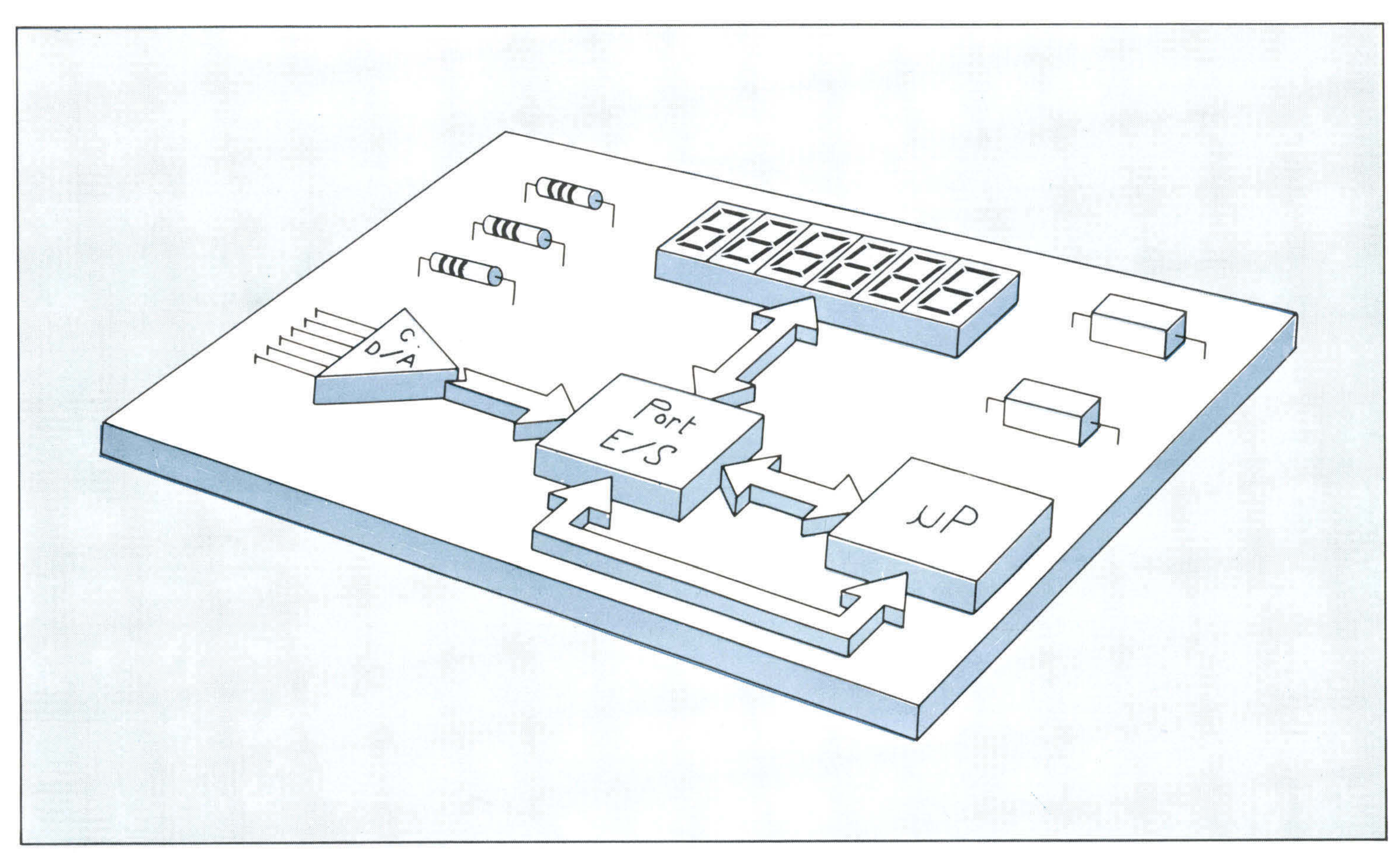
(d) Demander à un informaticien professionnel (si vous en avez un sous la main) sa définition du « temps réel ». ■

Jean-Michel COUR *

** Jean-Michel Cour anime le groupe « Architecture Informatique », spécialisé dans l'ingenierie micro-informatique au sein de la société CISI.*

Le μA 9708

Circuit d'acquisition de données compatibles avec les microprocesseurs 8 bits



Le μA 9708 de Fairchild* est un nouveau sous-ensemble économique, convertisseur analogique numérique de 8 bits à 6 canaux, conçu pour être utilisé avec les microprocesseurs MOS tels que le F8, le 3870 et le 6800.

Il réalise une conversion 8 ± 1/2 bits en 300 μs avec zéro automatique, correction de pleine échelle et possibilité de conversion ratiométrique.

Ce circuit utilise le microprocesseur pour la génération de l'adressage, des rythmes et du comptage. Il comporte un décodeur 1 parmi 8, un multiplexeur à 8 canaux, un échantillonneur bloqueur et un comparateur de sortie, le tout intégré dans un même boîtier enfichable en ligne standard, plastique ou céramique, à 16 sorties.

** FAIRCHILD : 121, avenue d'Italie, 75013 Paris. Tél. : (16.1) 584.55.66.*

Etalonnage

Lorsque le système nécessite un étalonnage, les adresses 000 et 111 de sélection de lignes peuvent être utilisées conjointement avec les possibilités de calcul du microprocesseur pour introduire le 0 volt et initialiser le facteur d'échelle.

Ainsi l'adresse 000 appliquée sur les broches A_0, A_1, A_2 connecte, à l'entrée du générateur de rampe, une masse pouvant servir de correction d'offset lors des conversions ultérieures. Quant à elle, l'adresse 111 place à l'entrée du générateur de rampe la tension de référence V_{REF} qui sera utilisée pour déterminer le facteur d'échelle.

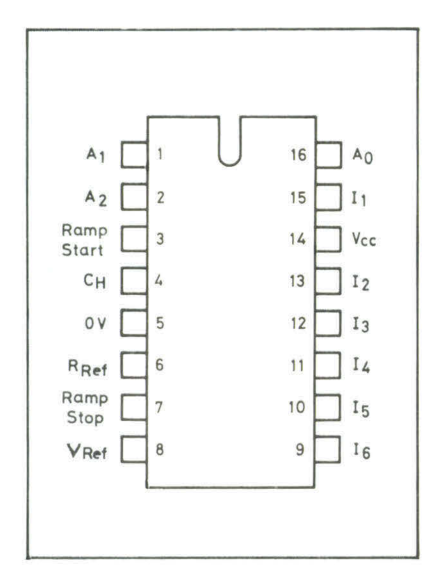
Choix des entrées

Six tensions analogiques différentes provenant de six détecteurs peuvent être converties en information binaire. Le choix d'une de ces entrées est déterminé, par le code binaire, appliqué sur les lignes d'adresse A_0, A_1, A_2 comme indiqué dans le tableau ci-dessous.

Ligne d'adresse			Entrée analogique sélectionnée
A_2	A_1	A_0	
0	0	0	0 volt
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	V_{REF}

Le temps nécessaire à la charge du condensateur C_H est le temps d'acquisition des données.

Brochage du convertisseur analogique/digital μA 9708 (vue de dessus).



Fonctionnement

Lorsque l'entrée « **Ramp Start** » ou démarrage de rampe est au niveau logique 0, la tension analogique issue des entrées I_1, I_2... ou I_6 est transférée à un condensateur extérieur C_H relié à la broche 4 du circuit **(fig. 1).**

Le temps nécessaire à la charge de C_H est le temps d'acquisition des données, lequel sera fonction de l'impédance de sortie de l'amplificateur interne du convertisseur et de la valeur de la capacité.

Lorsque C_H est chargé, on place un « 1 » logique sur l'entrée « Ramp Start » ce qui isole provisoirement (état haute impédance) la tension analogique d'entrée du condensateur.

Dès lors, C_H commence à se décharger à un rythme contrôlé. La pente de la rampe est déterminée par la tension de référence extérieure (V_{REF}), la résistance de référence (R_{REF}), la valeur du condensateur C_H et les fuites inhérentes au convertisseur.

Le condensateur C_H est relié à l'entrée d'un comparateur dont la sortie « **Ramp Stop** » (broche 7) permet l'arrêt du comptage. La sortie du comparateur est au niveau logique « 1 » lorsque le condensateur est chargé et au niveau logique « 0 » lorsqu'il est déchargé.

Fig. 1. – Bloc diagramme du convertisseur. Les 3 lignes d'adresse A_0, A_1, A_2, permettent de sélectionner une des 6 entrées I_1, ... I_6, la tension de référence Vréf. ou le 0 V pour la correction d'offset.

Fig. 2. - Diagramme des temps :
— t_A représente le temps nécessaire à la charge de C_H, c'est le temps d'acquisition des données ;
— t_R correspond à la durée de la rampe ou temps de conversion.

La durée de la rampe est le temps pendant lequel celle-ci passe du niveau 1 au niveau 0 **(fig. 2).** Le microprocesseur doit être programmé pour déterminer par comptage ce temps appelé aussi temps de conversion.

Ainsi, le temps de conversion idéal peut être calculé de la façon suivante :

Durée de la rampe ou temps de conversion $= V_1 \cdot \frac{C_H}{I_R}$

Avec
V_1 : tension analogique d'entrée à mesurer
C_H : condensateur extérieur
$I_R = \frac{V_{CC} - V_{REF}}{R_{REF}}$
V_{CC} : tension d'alimentation
V_{REF} : tension de référence
R_{REF} : résistance de référence.

La **figure 3** représente la fonction idéale de transfert du comptage des impulsions en fonction de la tension d'entrée lorsqu'il n'y a pas de correction d'offset ni d'erreur d'échelle (la pleine échelle correspond au cas où la tension d'entrée V_{in} est égale à la tension de référence V_{REF} pour un nombre de 256 impulsions comptées).

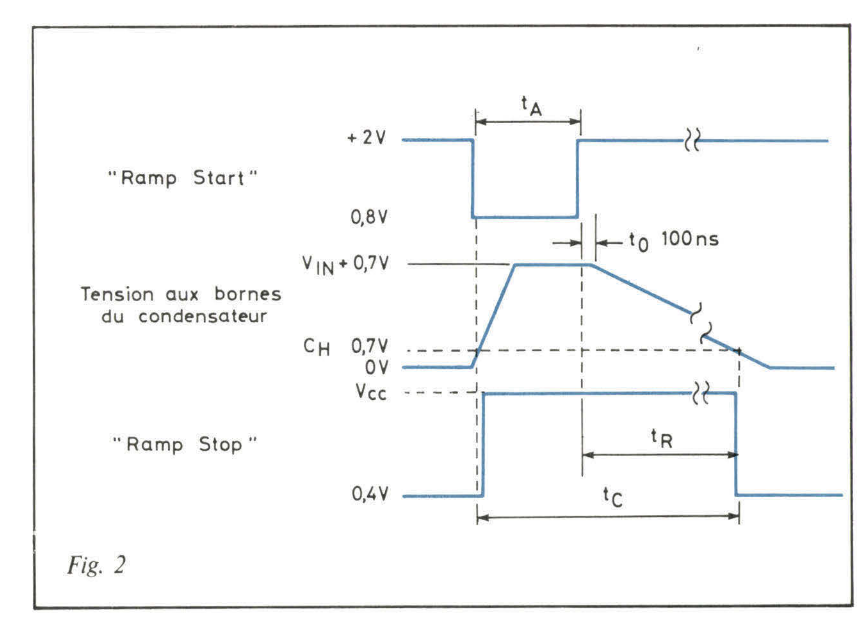


Fig. 2

Application

Un exemple d'application utilisant un convertisseur analogique digital du type μA 9708 est donné **figure 4.** Il s'agit de contrôler par microprocesseur l'opacité d'un liquide ou d'une solution circulant à travers un tube transparent.

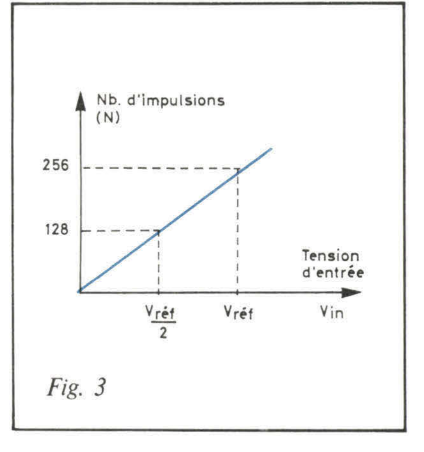


Fig. 3

Fig. 3. - Fonction de transfert idéale. Lorsque la tension d'entrée V_{in} est égale à Vréf., le microprocesseur doit compter 256 impulsions. L'équation est de la forme y = a x ou ici, en fonction de V_{in} :

$$N = \frac{256}{Vréf.} \cdot V_{in}$$

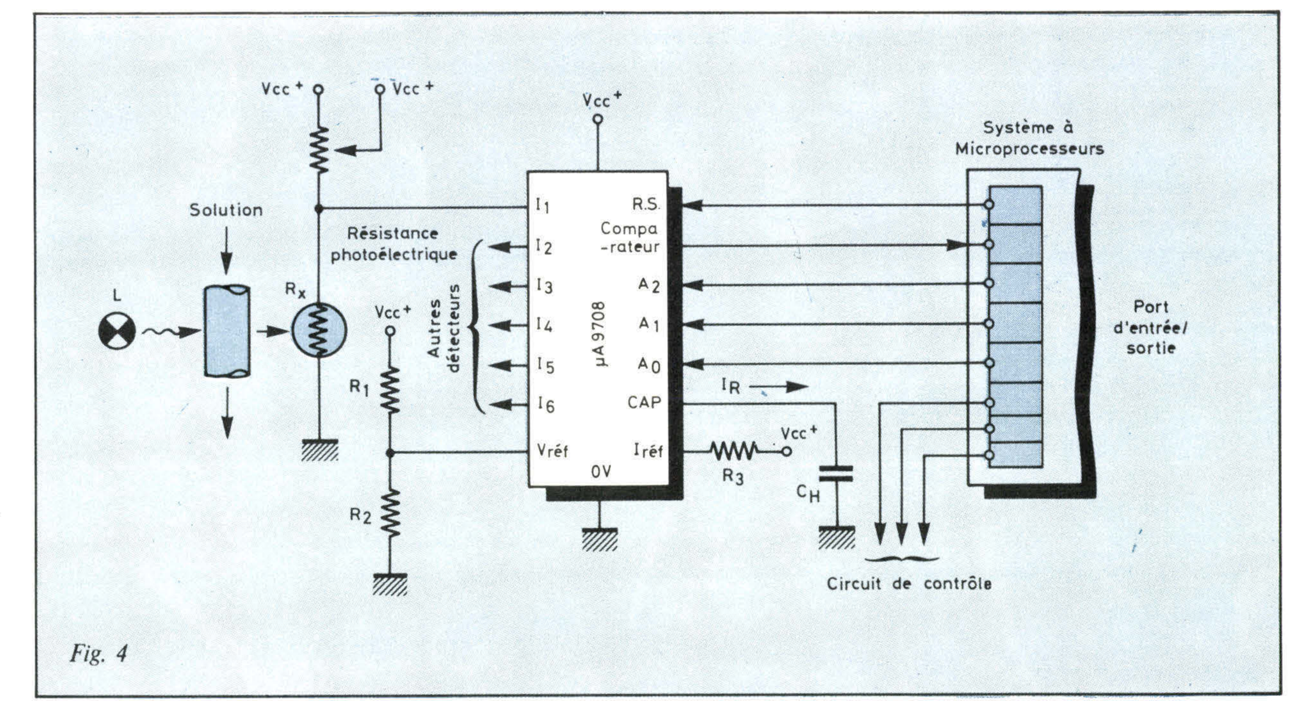


Fig. 4

Fig. 4. - Contrôle automatique par microprocesseur de l'opacité d'une solution. Le micro-ordinateur fournit aux circuits de contrôle les données leur permettant d'ajuster les composantes des différents mélanges mesurés.

L'intensité lumineuse issue de la lampe L est convertie par la photorésistance R_X en une tension, plus ou moins élevée suivant l'opacité du liquide.

Le convertisseur A/D délivre un créneau de longueur proportionnel à cette tension. Le microprocesseur compte le nombre d'impulsion contenu dans ce créneau et analyse le résultat du comptage. En fonction de ce résultat, il fournira aux circuits de contrôle, par l'intermédiaire du port d'entrée/sortie, les informations qui leur permettront d'ajuster l'opacité de la solution en jouant sur les composants du mélange. Le domaine d'applications de ce système concerne plus particulièrement le contrôle automatique de l'homogénéité des solutions chimiques et des mélanges. ■

Caractéristiques du μA 9708

- Compatible avec une unité centrale.
- Faible coût.
- Une seule tension d'alimentation.
- Linéarité : ± 0,2 % max., sur toute la gamme de température.
- 300 μs de temps de conversion typique par canal.
- Possibilité de zéro automatique et de correction de pleine échelle.
- Réglage du facteur d'échelle, les références de précision coûteuses ne sont pas nécessaires.
- Compatible broche à broche avec des modèles déjà populaires : MC 14443 et MC 14447.

Réalisez votre micro-ordinateur « Micro-Systèmes I »

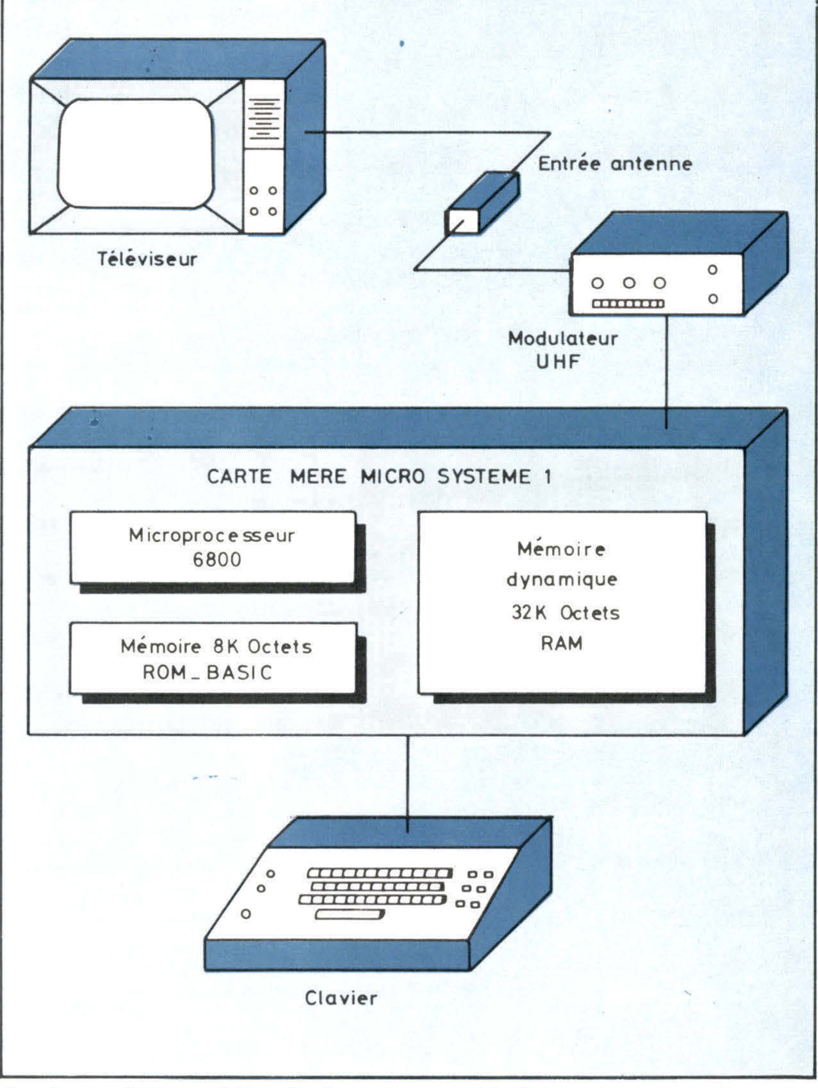
Outre le dialogue permanent que la revue veut instaurer entre ses lecteurs et les constructeurs, la vocation principale de Micro-Systèmes est de vous proposer et de vous assister dans la réalisation complète de micro-ordinateurs dont nous avons préalablement testé les performances. Entendons-nous sur la sémantique : certains considèrent comme « micro-ordinateur » tout dispositif capable de faire fonctionner des programmes, même s'il ne dispose que d'un langage-machine. Partant de là, tous les kits d'évaluation et systèmes de mise au point pourraient répondre à cette définition.

En fait cette appellation ne doit recouvrir que des machines disposant d'un langage évolué donc d'une mémoire vive capable de contenir des compilateurs ou interpréteurs de ces langages.

Lorsqu'à présent nous vous proposons la réalisation d'un micro-ordinateur, c'est à cette deuxième définition que nous nous référons. Nous en avons confié l'étude au Groupe Protéus International qui durant trois mois s'est appliqué à le rendre simple mais performant, facilement exécutable car exempt de tout réglage.

Ce micro-ordinateur est programmable en « Extented-BASIC », langage cher aux amateurs, car il s'apprend en quelques heures.

Un BASIC tel que celui que comporte « micro-systèmes 1 » permet les entrées/sorties en huit bits parallèles et la manipulation binaire d'appareillage électronique, les calculs à neuf décimales, le traitement de chaînes de caractères et de textes *, les appels de sous-programmes écrits par l'utilisateur directement en langage-machine, la programmation de jeux, la gestion, etc.



Le micro-ordinateur « Micro-Systèmes 1 » permet le dialogue homme-machine grâce à un clavier ASCII et à un téléviseur domestique.

Comment réaliser son micro-ordinateur ?

A partir du moment où vous décidez de vous initier aux microprocesseurs, la construction d'un micro-ordinateur devient presque une finalité vers laquelle chaque lecteur a de bonnes raisons de vouloir prétendre.

Est-il possible de se lancer dans cette aventure seul ? Regardons ensemble les étapes qui mènent à l'aboutissement de ce projet.

Premier scénario

Donc, vous êtes chez vous, tout seul, la micro-informatique vous passionne et vous aimeriez avoir votre micro-ordinateur construit selon votre étude personnelle.

La lecture de Micro-Systèmes vous a maintenant donné toute la confiance dont vous aviez besoin pour faire de ce rêve une réalité.

Vous prenez une feuille de papier, un crayon et votre matière grise commence à vous dicter les fonctions que devra comprendre votre système. De là vous recherchez autour de quel microprocesseur il sera organisé. A cela vous ajoutez quelques mémoires, un clavier, un affichage sur écran vidéo et en option un lecteur de cassette. Jusqu'ici tout va bien.

Les composants étant sélectionnés, il vous reste à étudier l'implantation de la carte-mère. Ça devient difficile mais à force de patience on peut s'en sortir. Elle sera nécessairement du type double face à trous métallisés.

Pour l'approvisionnement de vos composants, nous vous faisons confiance. Vous pourrez toujours trouver un endroit où les acheter au meilleur prix. Mais attention, plutôt que de privilégier les composants bon marché, il est de loin préférable, pour ce type de réalisation, de rechercher les composants de qualité.

A présent, ça y est tout est là. C'est gagné...

Pas tout à fait, car pour en arriver à ce stade nous venons d'élu-

** Une chaîne de caractères et de textes représente un nom, un mot code, une phrase que la machine traite sous forme de suite d'octets binaires.*

der un certain nombre de problèmes difficiles à surmonter. Lesquels ?

— La première grande difficulté apparaît dans l'étude de l'implantation de la carte-mère : faisable mais très laborieux.

— Deuxièmement sa réalisation pratique n'est pas envisageable avec les moyens dont dispose un amateur, de plus fabriquée à l'unité elle devient un composant de luxe.

Il faut ajouter à cela que de sa bonne exécution, dépend largement le fonctionnement de votre micro-ordinateur.

— Troisièmement, pour mettre au point un langage BASIC il faut pouvoir disposer d'un ordinateur du type IBM 360.

— Devant tous vos efforts, un quatrième problème va surgir : comment programmer la mémoire contenant l'interpréteur BASIC ?

Il faut d'abord définir le programme pour ensuite le stocker en mémoire.

Il est évident que ce scénario se terminera bien tristement.

Regardons le deuxième scénario

C'est simple, l'idée, votre idée, reste la même, mais à présent nous avons décidé de vous venir en aide avec « micro-systèmes I ».

« micro-systèmes I » n'est pas un kit.
« micro-systèmes I » est une **étude personnelle** assistée.

Que comporte cette assistance ?

a) La première chose que nous vous offrons c'est l'étude, nous l'avons faite faire pour vous par Protéus International.

Vous découvrirez la notice de ce système tout au long de cette rubrique.

b) La carte-mère : spécialement réalisée pour « micro-systèmes I », la revue sera le dépositaire exclusif de ce composant passif qui, marqué de l'estampille « micro-systèmes I », vous sera cédé au prix coûtant de 300 F (TTC).

Comportant au total 1400 trous métallisés, cette carte a été réalisée selon des normes professionnelles avec comme objectif fondamental, l'obtention d'un taux de déchet nul. Pour parvenir à ce résultat l'implantation a été confiée à un ordinateur...

Ainsi à l'image des matériels professionnels, cette carte est conçue pour supporter l'ensemble des composants nécessaires, excepté le modulateur UHF en option 44 F (TTC) (signal vidéo-complexe sur la carte-mère) et le clavier ASCII encodé. Type 1648 de Key-Tronic.

Cette carte reçoit les huit bits parallèles provenant de ce clavier et fournit à un poste de télévision équipé d'un modulateur UHF, ou disposant d'une entrée vidéo, le signal nécessaire à la visualisation de 16 lignes de 64 caractères de texte ou graphiques.

c) Une mémoire morte ROM de 64 K bits (8 K - octets) contenant le BASIC, dont la fabrication relève des toutes dernières techniques d'intégration et pour laquelle le prix de vente est de 980 F (TTC).

Cette mémoire pourra être remplacée par quatre mémoires préprogrammées d'EPROM'S.2716 (2 K octets chacune) disposées sur une plaque de circuit imprimé terminé par un connecteur de 24 broches pouvant s'insérer dans le support réservé à la ROM-BASIC.

Cet ensemble de mémoires préprogrammées EPROM'S est disponible au même prix que la ROM-BASIC « micro-systèmes I ».

Toutefois, il est à remarquer que cette dernière s'alimente uniquement en + 5 V alors que la batterie d'EPROM'S nécessite les tensions suivantes : + 5 V ; - 5 V ; + 12 V.

Maintenant que les quelques points, qui jusque-là demeuraient difficilement accessibles même aux amateurs chevronnés, ont trouvé une solution efficace et économique, il ne reste plus qu'à passer aux choses sérieuses.

Le reste des composants est classique et possède plusieurs sources d'approvisionnement.

Pour les mémoires dynamiques, il a été choisi des SV6726 pour leur grande rapidité. D'une capacité de 16 K × 1 bits, elles seront montées par quantité correspondant à des multiples de huit.

Le prix de l'ensemble des composants se situera dans la fourchette des 3000 à 4000 F (TTC) pour la version complète qui correspond à un micro-ordinateur dont les performances le placent dans le peloton de tête des systèmes actuellement disponibles sur le marché français.

« Micro-systèmes I » fonde sa supériorité sur la puissance de son langage et sur sa puissance de calcul.

Votre investissement initial pourra se porter sur l'acquisition de la carte-mère, tous les autres éléments pouvant être achetés au fur et à mesure de la progression de votre étude « micro-systèmes I ».

Pour la mémoire BASIC, qu'il s'agisse de la mémoire ROM ou de la batterie d'EPROM'S 2716 préprogrammées, ces deux composants constituent des fournitures spécifiques approvisionnées par Protéus International.

Un micro-ordinateur ! Pour quoi faire ?

Si dans un premier temps la motivation qui vous anime est l'étude et la réalisation de « micro-systèmes I », il n'en demeure pas moins vrai que ceci ne constitue pas une fin en soi et qu'en fait ce que vous visez réellement c'est l'utilisation de ce micro-ordinateur en réponse à des problèmes spécifiques ou généraux que vous rencontrez.

Certains dirons : « oui mais pourquoi un micro-ordinateur plutôt qu'un système d'évaluation ou de mise au point de microprocesseurs travaillant en langage machine ou assembleur ? » Tout simplement parce qu'il les remplace avantageusement.

En électronique

A part les applications militaires ou industrielles, où le problème du gain, recherché au niveau de la microseconde se pose avec acuité un utilisateur disposant d'un peu de temps pour son application (il s'agit ici d'une vue de l'esprit car ce temps est microscopique et imper-

ceptible à l'échelle humaine) aurait avantage à calculer les paramètres de son expérience et à les programmer en langage évolué.

Ceci est concevable pourvu que la liaison avec les périphériques soit assurée comme dans tous les systèmes à microprocesseurs et soit accessible par le langage.

Prenons un exemple pour mieux comprendre : un ingénieur doit calculer « U » (tension crête), « I » (courant crête) dans une expérience sur les circuits fonctionnant en impulsions.

Il lui faut d'abord multiplier des données binaires de huit bits issues d'un convertisseur analogique-digital par des facteurs d'échelle pour ensuite afficher la valeur de l'impédance Z = U/I.

En plus de ces multiplications et de cette division binaire, les résultats doivent être affichés en décimal d'où une conversion finale Binaire-BCD.

Pour ce faire, il commence par travailler en binaire sur un kit d'initiation. A mi-chemin dans son projet, une discussion le met sur la voie d'un micro-ordinateur du type de celui que nous vous proposons d'étudier.

Ainsi, après l'acquisition de ce micro-ordinateur, son application se traite en quelques heures : avec saisie des données en binaire, calculs en BASIC et affichage sur écran vidéo au lieu d'afficheurs à sept segments. Ainsi dispose-t-il enfin d'un moyen **pratique, efficace et rapide** de travail, car la programmation et l'arithmétique hexadécimale binaires ne sont pas toujours des plus attrayantes

Dans la vie de tous les jours

« micro-systèmes I » constitue un véritable **outil informatique**. En effet, malgré le fait qu'il soit édifié sur des composants simples et limités en nombre, cet ordinateur ne mérite absolument pas ce préfixe au regard de sa puissance de calcul.

Il offre la possibilité de programmer : la gestion de son (ou ses) compte (s) en banque (s), les achats ménagers à effectuer pour la semaine à venir, des jeux éducatifs ou distractifs visualisés sur télévision, bref toutes ces applications qui ont déjà fait l'objet de nombreuses émissions télévisées.

En plus de l'aspect **« Jeu »,** sa programmation offre des possibilités de **« Gestion ».**

Ainsi « micro-systèmes I » est tout aussi bien destiné aux « PME » qui ont de bonnes raisons de ne pas investir dans la grosse informatique compte tenu de la dimension de leurs problèmes. (On ne prend pas un marteau-pilon pour briser une noix).

Pour une mise de fonds minimum, il gère toute une comptabilité et peut assurer un travail de secrétariat tel que l'impression de **courrier** répétitif, remplaçant de la sorte certaines machines de gestion de textes très onéreuses. Il vous apporte la solution que vous attendiez pour composer automatiquement vos **numéros de téléphone,** il peut même, après avoir obtenu votre correspondant transmettre du courrier, des données, etc.

Ce micro-ordinateur peut tenir une **caisse,** traiter des **calculs** qu'on lui soumet en un langage clair et dont tout le déroulement est suivi en permanence sur l'écran de votre téléviseur.

Mais attention, ceci n'est en rien comparable avec les calculatrices programmables les plus sophistiquées.

Ajoutons à tout cela qu'il est capable d'imprimer ses résultats sous forme de tableaux ou sous forme de graphique si on le lui demande.

A qui nous adressons-nous ?

Nos efforts s'adressent à la grande majorité, enthousiaste et à la fois surprise par l'ampleur de ce phénomène qui conduit à une démocratisation de l'informatique.

La construction du « micro-systèmes I » pourrait être perçue comme un simple exercice de câblage nécessitant un fer à souder, de la soudure, des composants et le circuit imprimé double face à trous métallisés. Certes, il est permis de s'en tenir à cette solution pour qui n'est pas familiarisé avec l'électronique, mais rappelons qu'il s'agit d'abord d'entreprendre l'étude de ce micro-ordinateur pour lequel nous serons là pour vous assister.

D'un autre côté être ingénieur électronicien ne constitue pas une nécessité car beaucoup plus que la réponse transitoire en impulsions des composants, il importe de se soucier de leur sens d'implantation.

La récompense finale de tout votre travail sera matérialisée par l'apparition sur l'écran de votre téléviseur du « Ready » du BASIC.

Quant à la notice de montage qui commencera dans le numéro prochain, elle sera accompagnée des principes de fonctionnement, des particularités du micro-processeur et des mémoires utilisées dans ce système.

Ce qui veut dire, qu'à l'instar de l'équipe qui a conçu « micro-systèmes I », vous découvrirez les raisons qui ont prévalu dans leur choix ainsi que la justification de l'architecture de ce micro-ordinateur.

En résumé, nous traiterons de :

A : la présentation de la fiche technique du micro-ordinateur.

B : la description de son organisation générale accompagnée d'une analyse du matériel (hardware).

C : la réalisation pratique de la carte-mère.

D : des explications détaillées sur le fonctionnement de ce micro-ordinateur ainsi que la réponse aux questions suscitées par son montage.

E : la programmation de nombreuses applications avec des programmes en Basic.

A ce sujet le « volume de la sphère », la « visualisation de courbes par points » qui vous ont été présentées précédemment au chapitre « programmation » sont des programmes qui « tournent » déjà sur « micro-systèmes I ».

Fiche technique de « MICRO-SYSTEMES I »

HARDWARE

Présentation :

Une plaque-mère, reliée à un clavier encodé ASCII et un écran TV.

Processeur :

6 800 (SFF 96 800) ; 0,894 MHz (2 à 4 cycles/instruction en moyenne).

Rom :

Standard BASIC-8k résident.
Option assembleur, éditeur de texte sur cartes enfichables.

Ram :

32 k octets sur supports, soudée sur la plaque-mère.
16 k octets en option sur des cartes enfichables.

Entrées/Sorties :

Standard TTY, entrée/sortie série TTL, en boucle de courant 20 mA et standard RS 232, trois versions résidentes en permanence sur la plaque-mère.
Minicassette en standard Kansas City (1 200 Hz × 4 cycles pour un zéro et 2 400 Hz × 8 cycles pour un « 1 » logique). **Options** 4 cartes enfichables sur supports bénéficiant des bus tamponnés ainsi que d'un décodage d'entrée/sortie, permettant d'installer des PIA-s.

Option Floppy Disk :

Un contrôleur de DMA sur la plaque-mère et une carte sur support permettent de gérer 3 minidisquettes Pertec double face et double densité (2,6 Mbits de capacité totale chacune). DOS chargé en RAM (16 k octets) à partir de la disquette.

Alimentation :

+ 5 V ± 10 % ; 3 A
+ 12 V ± 10 % ; 1,5 A
− 12 V ± 10 % ; 1 A

Non fournie. Une alimentation résidente sur la plaque, dérivée de − 12 V.

Clavier ASCII :

Un côté « B » de PIA (PB_0-PB_7, CB_1, CB_2) reçoit les 7 bits, un « Strobe » reçu sur le front descendant, un « Break » et « Here Is », d'un clavier encodé ASCII-standard.

Ecran Vidéo

La plaque fournit un signal Vidéo-Complex pouvant gérer un moniteur vidéo ou un écran TV ordinaire muni d'un modulateur U.H.F.
Le circuit de visualisation de caractères graphiques, résident sur la plaque possède sa propre mémoire d'écran. Les caractères sont « fondus » dans une EPROM-2708 et peuvent être modifiés à souhait par l'utilisateur. La version standard contient le jeu de caractères ASCII.

SOFTWARE

Structure de l'espace adressable

Les divers éléments occupent les adresses indiquées dans le **tableau I**.
On constate une occupation continue des premiers 32 k octets, de OOOO à 7 FFF. Ces adresses contiennent la RAM dynamique logée intégralement sur la carte-mère.
Les 16 k octets suivants, allant de 8 000 à BFFF sont réservés à un coupleur d'entrée/sortie parallèle (le « PIA »-MC 6820) et série (l'« ACIA »-MC 6850) et à la mémoire où les coupleurs d'entrée/sortie additionnels qui prendront place sur des cartes enfichables.

BASIC

La mémoire ROM-8 k octets contient un BASIC étendu permettant toutes les fonctions du FULL-BASIC plus des fonctions spéciales. Nous les avons résumées dans le **tableau II.**

Tableau I

FFFF EOOO	ROM BASIC ou Amorce pour le chargement du DOS
DFFF COOO	8 k-ROM BASIC
BFFF 8000	LIBRE 8010-8011 TTY, imprimante-ACIA 8004-8007 clavier 8 vidéo-PIA
7FFF OOO	32 k de RAM sur la plaque-mère

TABLEAU II

Le jeu d'instructions BASIC de MICRO-SYSTEMES I

Commandes :

RUN : exécution immédiate
NEW : efface la mémoire de programme
CONT : après un STOP, permet de continuer
TRACE ON : démarre l'exécution en pas à pas en BASIC
TRACE OFF : arrête le pas à pas
FORT : définit le port (canal, ou ligne) d'E/S
LOAD : charge le contenu d'une cassette, disquette, etc. en mémoire.
APPEND : comme LOAD sauf qu'il n'efface pas (NEW) l'ancienne zone pgme
SAVE : sauvetage d'un pgme sur cassette, ruban perforé ou disquette
ER : rend la main au moniteur, DOS, etc. s'ils existent
LIST : affichage de la liste d'instructions du programme
INCR : numérotage automatique par pas de INCR des lignes
LINE : permet de réserver une ligne de ligne caractères
DIGITS : choisit le nombre de décimales désirées
SPEED : choix, par programme de la fréquence d'horloge-système
STRING : introduction d'une chaîne de caractères (plus de 18)
CTRL C : STOP de l'exécution
CTRL X : effacement d'une ligne en cours d'entrée
CTRL O : effacement du dernier caractère tapé
RUBOUT : numérotation automatique

Directives :

REM : remarque-ligne non-exécutable, de commentaire
LET : « soit » affectation (facultative) LET X = Y + Z
PRINT : « imprimer » utilisé pour l'impression de texte et résultats
INPUT : « entrer » utilisé pour l'entrée de texte et données
GOTO : « va en » saut inconditionnel
END : fin du programme
STOP : arrêt
IF THEN : saut logique ou exécution conditionnelle
FOR : début de boucle
NEXT : fin de boucle
STEP : pas de boucle
GOSUB : saut à un sous-programme
RETURN : retour d'un sous-programme
ON GO : aiguillage inconditionnel
ON GOSUB : aiguillage vers des sous-programmes
DIM : tableau
DATA : introduction de données
READ : lecture des données précédentes
RESTORE : manipulation du pointeur du READ
DEF : définition des fonctions propres du programmeur

Fonctions :

SQR : racine carrée SQR (9) = $\sqrt{9}$
ABS : valeur absolue
INT : valeur entière
SGN : signe : positif = **1** ; négatif = **– 1** ; nul = **0**
RND : produit des nombres aléatoires
DLOG : logarithme décimal
LOG : logarithme naturel
EXP : exponentielle
SIN : sinus
COS : cosinus
TAN : tangente
ATAN : arctangente
DEG : conversion radians-degrés
RAD : conversion degrés-radians
TAB : tabulation - déplace le pointeur
POS : position du pointeur
LEN : longueur d'une chaîne de caractères
VAL : valeur du nombre = chaîne de caractères
STR$: variable numérique dans une chaîne de caractères
ASC : donne la valeur décimale d'un caractère
CHR$: chargement d'un caractère
LEFT$: nombre de caractères, partant de la gauche
MID$: nombre de caractères, au milieu
RIGHT$: nombre de caractères, partant de la droite
PEEK : lecture d'un mot mémoire (en décimal)
POKE : écriture dans une case mémoire
USEP : appel de sous-programmes en langage machine
FN : création de fonctions

Organisation générale

Générateur d'horloge

Le synoptique de la **figure 1** donne une idée de l'architecture d'un système micro-ordinateur classique. L'unité Centrale représente bien peu de chose. Elle reçoit ses signaux d'horloge de la part d'un circuit qui génère en même temps les signaux garantissant le D.M.A. (Direct Memory Access), le rafraîchissement de la mémoire dynamique et crée la fréquence d'horloge d'un diviseur-générateur de bauds, nécessaire aux transmissions série.

Les amplificateurs de ligne

Le synoptique ne fait pas apparaître les amplificateurs, ou « tampons » de ligne dont est munie l'Unité Centrale.

Ses bus de données et d'adresses ainsi amplifiés, peuvent supporter une bonne dizaine de charges TTL, circuits sur la carte-mère, ou sur les supports des cartes enfichables de l'utilisateur.

La ROM-BASIC

L'Unité Centrale est reliée à une mémoire de programme-ROM qui contient en codes-machine les programmes nécessaires à la gestion de la carte (sortie vers un magnétophone mini-cassette ou TTY, entrées d'interruption, ou de données, etc.) ainsi que les programmes de calcul et d'interprétation des lignes BASIC. Au niveau du synoptique, certains codes-parallèle huit bits arrivant du coupleur PIA, ou codes-série venant de l'ACIA *, constituant autant de caractères de la syntaxe du BASIC ou des chiffres en décimal, sont stockés, ligne par ligne. Une ligne représente pour la machine ni plus ni moins qu'une suite de caractères ASCII, représentés en binaire et se terminant par le code « Retour Chariot » (RC). Le programme interpréteur BASIC fait entrer en mémoire un caractère après l'autre, tout en veil-

* *ACIA : Asynchronous Communications Interface Adapter.*

lant s'il n'est pas le « R.C. ». A l'arrivée de cet ordre (matérialisé par l'encodage sur le clavier d'une combinaison binaire comme les autres), le programme effectue une recherche syntaxique et sémantique de la chaîne de caractères reçue et s'il retrouve un ordre connu, il l'exécute. Par exemple, si l'on envoie l'ordre BASIC-direct :

GO TO 20 C.R.

c'est-à-dire la suite d'octets (correspondant au code ASCII) :

G = 79_{10} = 4 F_{16}
O = 71_{10} = 47_{16}
blanc = 32_{10} = 20_{16}
T = 84_{10} = 54_{16}
O = 71_{10} = 47_{16}
blanc = 32_{10} = 20_{16}
CR = 10_{10} = OA_{16}

Les choses se passeront de la manière suivante

Le « Strobe » généré par le clavier à chaque enfoncement de touche fera saisir par la PIA les huit bits parallèles que lui présente l'encodage du clavier, cet octet sera comparé à l'encodage du code représentant le retour chariot et sera rangé dans une mémoire de ligne. Ainsi cette mémoire tiendra sur plusieurs mots consécutifs : 47, 4F,...

A l'arrivée du Retour Chariot, le programme commence à analyser le premier caractère, puis les deux premiers caractères, etc., et les compare avec une table : Est-ce un « RUN » ?, un « FOR », un... etc. En trouvant « GO TO » qui est syntaxiquement correct à l'insu d'un « GO TA » par exemple, il branchera le processeur au programme de saut inconditionnel qui contiendra un code-machine.

BRA	Adresse relative (branchement)
ou	
JMP	Adresse absolue (saut)

Il est important de signaler que le temps passé à interpréter les ordres pour la gestion des codes (qui peuvent arriver par TTY, en série également), est incomparablement plus long que le temps d'exécution effective de l'instruction de saut. Néanmoins, un utilisateur humain est relativement insensible à une différence de temps d'exécution d'une dizaine de millionnièmes de seconde (BRA, JMP) ou d'un millième de seconde (GO TO).

Telles sont les grandes lignes du travail de la ROM-BASIC.

Le Contrôleur DMA

Pour faciliter les transferts rapides de données vers la mémoire, le système est muni d'un boîtier capable d'analyser les instructions présentes sur les bus et le cycle en cours. Ce boîtier d'accès direct à la mémoire est référencé DMA (Direct Memory Access). Les bus du microprocesseur se trouvent dans leur état actif durant les brefs moments, puis sont mis à l'état haute-impédance. Les rapports des temps de bus actifs/inactifs diffèrent d'une instruction à l'autre. Quoi qu'il en soit, pendant que le microprocesseur n'utilise pas ses bus, un autre processeur pourrait se servir de ses ressources (ROM-RAM, coupleurs, etc.) Mise à part les configurations multi-processeurs, la technique de l'accès direct à la mémoire est utilisée lors de la manipulation d'informations stockées sur les disques magnétiques. La cadence de ces transferts étant incomparablement plus rapide que celle des canaux TTY mini-cassette ou autres, il est parfois impératif de charger ces informations en mémoire sans passer par la technique traditionnelle : la PIA, va interrompre le processeur, prendre possession du bus de données et transférer un octet dans l'Unité Centrale qui le rangera à l'adresse mémoire « X ».

Ce procédé demande 20 à 50 μs pour le rangement d'un seul octet alors que pendant ce même laps de temps, avec des mémoires faites pour des cadences du MHz on peut y ranger 20 à 25 octets lors d'un accès direct.

Les disques souples

Le système utilise un contrôleur de disques souples pouvant gérer trois minidisquettes double face et double densité. Ce sont des supports magnétiques sur lesquels les informations sont écrites bit par bit, sans signaux d'horloge, comme c'était le cas des disques à simple densité. En effet, dans les anciens systèmes, les bits de données se trouvaient situés dans l'intervalle entre deux tops d'horloge.

La présence d'un top au milieu signifirait un « 1 » son absence – un « 0 » logique. Par les techniques modernes, liées à l'emploi des boucles de phase, on peut se passer des tops d'horloge et synchroniser néanmoins le récepteur à l'aide de préambules du type 10101010.. etc., qui lancent les oscillateurs des boucles de phase à la bonne fréquence, avant l'arrivée du bloc de données. C'est un peu acrobatique, mais ça marche sans faille et l'on stocke deux fois plus d'informations sur les mêmes longueurs de piste magnétique du disque.

Le décriptage et l'écriture des informations sur le disque demandent non seulement de l'électronique, mais aussi des programmes, pour l'organisation des données en blocs, le calcul des parités et des codes correcteurs d'erreur, etc.

Ces programmes sont stockés dans une PROM que nous avons appelée AMORCE **(tableau 1).** Les anglo-saxons l'appellent **Bootstrap.** En initialisation, l'AMORCE permet de charger en mémoire un programme situé sur le support magnétique lui-même.

En particulier, elle charge un programme de gestion des disquettes : le D.O.S. (disk operating system). Ce programme permet de faire des recherches, des copies et transferts de fichiers, etc.

Le minicassette

Un autre support magnétique que peut utiliser le système est le magnétophone minicassette ordinaire.

Dans ce cas les informations binaires sortant en série du coupleur d'entrée/sortie série (ACIA), sont traduites en cycles sinusoïdaux, aptes à être enregistrées sur bande comme toute oscillation « musicale ». Pas besoin d'une électronique compliquée pour obtenir ce genre de formattage. Le standard Kansas City d'enregistrement utilisé, veut qu'un « 1 » logique soit représenté par huit cycles à 2400 Hz et un « 0 » par 4 cycles

figure 1

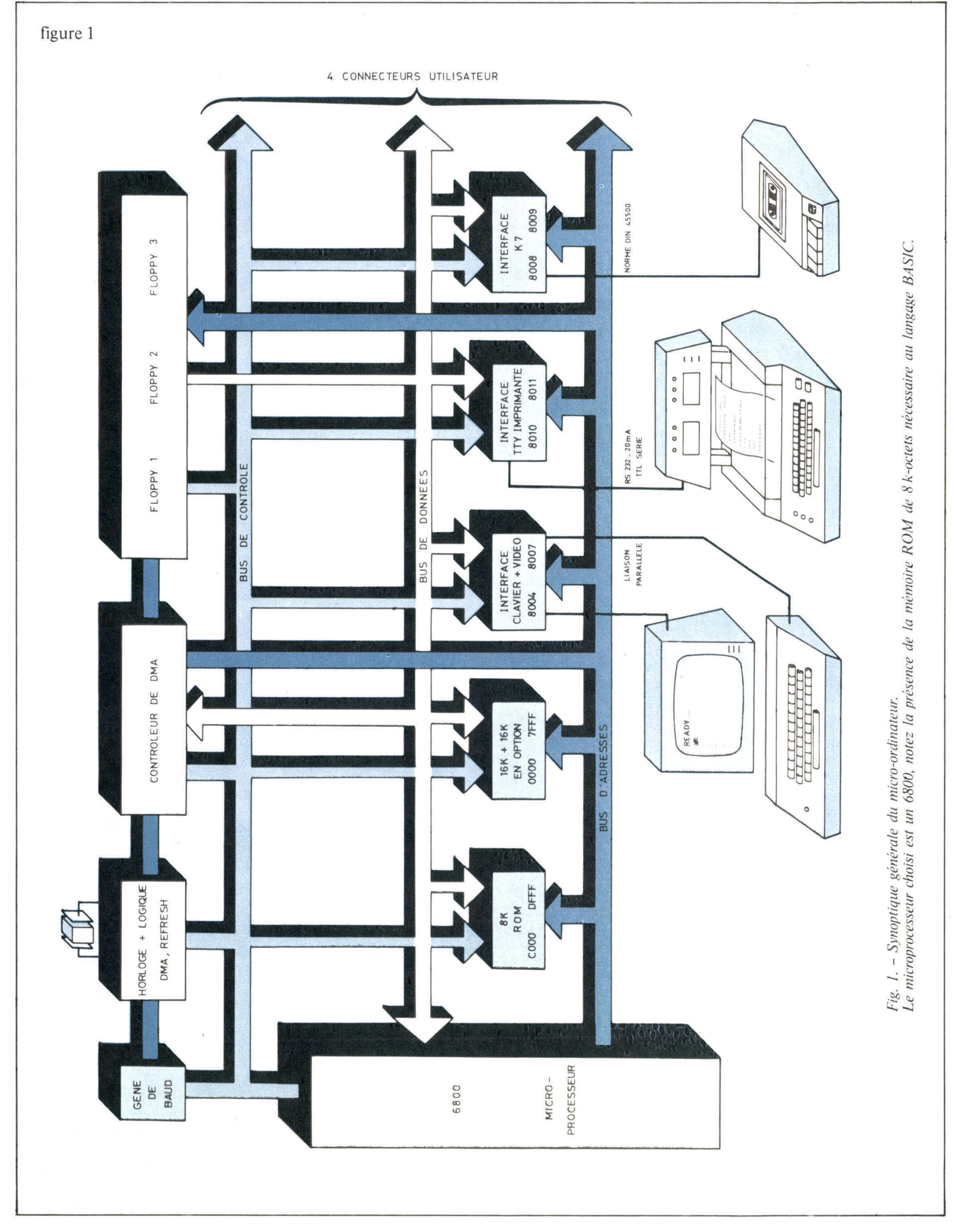


Fig. 1. – Synoptique générale du micro-ordinateur.
Le microprocesseur choisi est un 6800, notez la présence de la mémoire ROM de 8 k-octets nécessaire au langage BASIC.

de 1200 Hz. A la réception, deux filtres digitaux (à monostables) discriminent les fréquences enregistrées et les transforment en niveaux TTL logiques, envoyés dans l'entrée série de l'ACIA.

Les données à lire ou à enregistrer sont organisées en blocs de 256 octets séparés de portions de « 0 » logiques par programme, de sorte qu'un utilisateur peut avoir une idée du stade de l'enregistrement ou de la lecture en dénombrant les blocs. Le même programme organise les adresses d'enregistrement, en les situant en tête du bloc, etc.

La console de visualisation et clavier

Comme pour le magnétophone minicassette, la visualisation du texte (poste T.V. ordinaire muni d'un modulateur U.H.F. de jeux T.V., par exemple) et le clavier encodé ASCII sont pris en charge par le programme, contenu dans la ROM-BASIC. Le clavier peut fournir huit bits parallèles accompagnés de signaux de validation « strobe » ou d'arrêt « Break », etc.

La visualisation du texte sur l'écran utilise un boîtier « Contrôleur CRT » bien connu le SF.F 96364 de la Sescosem qui rafraîchit sur l'écran le contenu d'une mémoire de 1 k octets, selon le jeu de caractères se trouvant dans une EPROM-2708. Une microprogrammation permet à ce contrôleur d'écran cathodique de gérer le texte (passage à la ligne à l'arrivée d'un retour chariot, déplacements du curseur, saut de page, rouleau, etc.) à partir de codes-caractère qu'il reçoit en parallèle.

Les supports de cartes enfichables

Ils sont au nombre de quatre et restent à la disposition de l'utilisateur qui pourra y loger jusqu'à quatre cartes enfichables obtenues par câblage ou par wrapping et dont l'adjonction lui permettra de personnaliser son micro-ordinateur en fonction d'une application précise qu'il souhaite traiter.

Par exemple un aveugle aura toutes les facilités à utiliser « micro-systèmes I » s'il y ajoute une carte de réponse vocale et une carte d'entrée vocale.

Comment doit procéder l'industriel qui souhaite tenir tête à une raffinerie de pétrole ou automatiser un complexe industriel ? Très simple. Il lui suffit d'enficher 4 k octets d'adresses de PIA ce qui lui fait 1 000 PIA disponibles, chaque PIA pouvant être adressé et lu par le BASIC.

Ajoutons à cela qu'un PIA correspond à 16 fils d'entrée/sortie de travail, chacun de ces fils pouvant actionner une électrovanne par exemple : vous conviendrez qu'il devient aisé de s'offrir les grandes eaux de Versailles modulées par la musique d'un somptueux son et lumière.

André DORIS
Jean-José WANÈGUE

Le H 8

Le H 8 de Heathkit * a été mis au point en 1975 et fut l'un des premiers micro-ordinateurs à usage personnel introduit sur le marché. Ces années d'expérience portent leur fruit en ce qui concerne le logiciel : très riche et rodé, et sur la facilité d'emploi, car la machine est dotée d'innombrables commodités.

Son aspect est celui d'un mini-ordinateur classique, logé dans un rack dont la face avant forme un pupitre, à l'image du MITS-Altair, mais aussi des PDP-8 et 11 de Digital ou d'autres mini-ordinateurs professionnels. De structure variable, il ne réunit pas dans un même boîtier le clavier, la visu, l'unité centrale et les mémoires, comme certains modèles, mais permet, au contraire, à l'utilisateur de modifier à souhait la configuration de travail : unité centrale, imprimante à aiguilles DEC-Writer II, console de visualisation à clavier, par exemple, ou d'autres configurations pouvant inclure le double disque souple WH 17 ou un lecteur-perforateur de ruban H 10.

Outre l'expérience et la renommée de Heathkit dans le monde des amateurs, l'ensemble bénéficie de notices devant lesquelles nombre d'éditeurs palissent : les dessins expliquent le moindre geste, après chaque opération on biffe une case comme dans les check-list des gros avions...

Le logiciel a déjà la patine de l'utilisation : commodités à l'écriture des instructions. Pour taper l'instruction PRINT au clavier, par exemple, il suffit de frapper P, R, pour voir apparaître le reste de l'instruction (INT) automatiquement. Le texte sur l'écran peut être organisé en plusieurs colonnes. Rares sont les consoles de visualisation professionnelles disposant de telles facilités : possibilité d'avoir une vue d'ensemble d'un long programme. Ceci permet de se passer d'imprimante lors de la mise au point des programmes.

A l'image des mini-ordinateurs classiques, le rack-unité central possède des voyants indiquant le fonctionnement en interruptions (INT), la passation du contrôle au moniteur (MON), l'état de marche du processeur (RUN), l'alimentation secteur. Il possède un affichage 7 segments des bus d'adresse et de données, en octal, comme pour montrer son affinité pour les ordinateurs puissants.

Un clavier de commandes moniteur permet le travail en langage machine : adressage, lecture/écriture mémoire, sélection et visualisation des registres du processeur, lancement des programmes. Rappelons, pour finir cette présentation générale, qu'une politique commune avec Digital mène à une combinaison de ressources : certains terminaux communs et l'incorporation du LSI-11 dans l'ordinateur Heathkit H 11 en sont la preuve.

Structure

Le cœur du système est un microprocesseur 8080, possèdant 78 instructions, 7 registres internes et une vitesse maxi de fonctionnement de 2 MHz.

Il existe un programme-moniteur gérant le panneau avant, affichage en octal, signalisation et réception des commandes. Il occupe 1 k-octets de mémoire ROM à masque. La mémoire vive de toutes les versions standard est de 8 k-octets en RAM.

Le coffret micro-ordinateur est muni d'une plaque



Le système H 8, la console de visualisation H 9 et le lecteur, perforateur de ruban H 10.

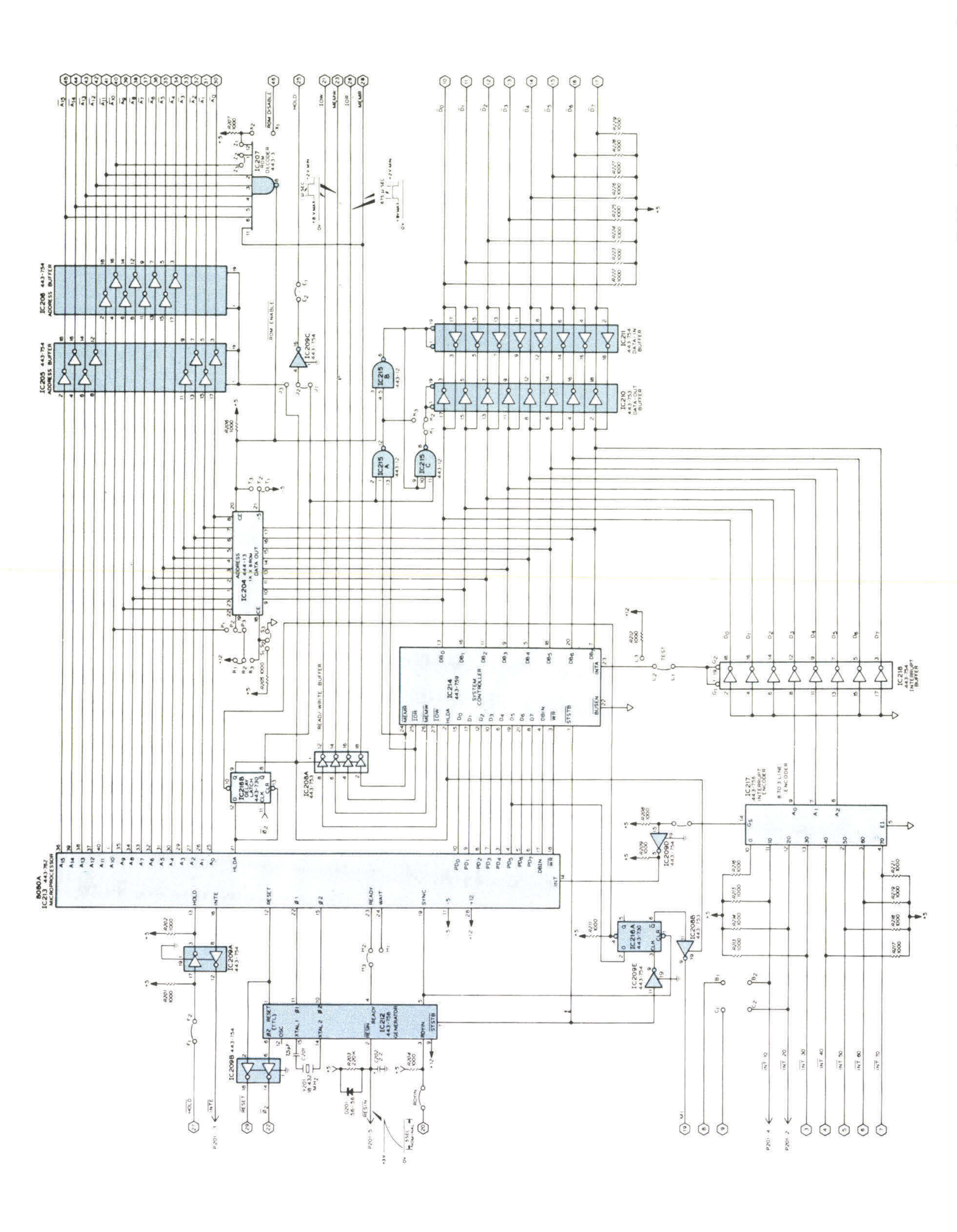


Schéma général du micro-ordinateur H 8.

mère possèdant un bus qui relie des supports, dans un standard propre (bus H 8), différent du bus S100. Néanmoins, étant donné que les deux bus servent le même type de microprocesseur, il n'est pas difficile de passer de l'un à l'autre. La différence majeure sur le bus H 8 comparé au bus S100, réside dans la structure du bus de données. Sur le S100 il y a 8 broches de données en sortie et 8 broches de données en entrée. Sur le bus H 8 les 8 broches de données peuvent être des entrées et des sorties, d'où une économie de broches. Le connecteur résultant est plus petit et le prix de revient de l'ensemble en est diminué.

Des liaisons séries sont présentes et permettent la connexion d'un mini-cassette, d'un terminal TTY (téléimprimeurs, ou machine à écrire reliée par une liaison-série) et d'un écran cathodique muni de clavier. Trois ports d'un 8255 - coupleur d'entrée/sortie de la famille INTEL, offrent, en option, des liaisons en parallèle - 8 bits avec possibilité de programmation en entrée et en sortie. Le H 8 est donc doté de toute la panoplie d'un mini-ordinateur professionnel, permettant son interconnexion dans un système plus vaste.

Programmation

Le choix du 8080 rend possible l'utilisation d'un logiciel très riche conçu autour de ce microprocesseur. La bibliothèque de programmes en Assembleur 8080 est peut-être la plus complète du monde aujourd'hui, ou, tout au moins, la plus populaire.

L'amateur dispose de langages évolués. Leurs interpréteurs ou compilateurs se logent en RAM. La carte mère peut recevoir des cartes mémoires permettant d'accroître les 8 k standard à 12 k, 16 k octets, etc.

Nous avons testé un Basic « B-H » (Benton-Harbor) dont la bibliothèque de programmes ne cesse de s'agrandir. Il fonctionne avec 12 k octets de RAM et même 16 k octets quand on utilise toutes ses fonctions.

En plus du Basic, le H 8 possède un Assembleur 8080 en 8 k octets référencé HASL-8 et un Editeur de Texte (TED-8), de 8 k octets également. Il existe en outre un programme-moniteur permettant de travailler à partir d'une console TTY ou CRT, appelé BUG-8 et logé en 3 k octets.

Ces programmes peuvent se charger à partir d'un mini-cassette ou de l'unité de disques souples et confèrent au H 8 la qualité de Station de Mise au Point de programmes 8080.

Conclusion

A la différence d'autres systèmes, le H 8 offre à l'utilisateur un choix d'architecture propre, universel, à la convenance des utilisateurs (programmeurs en Basic, électroniciens, automaticiens, etc.).

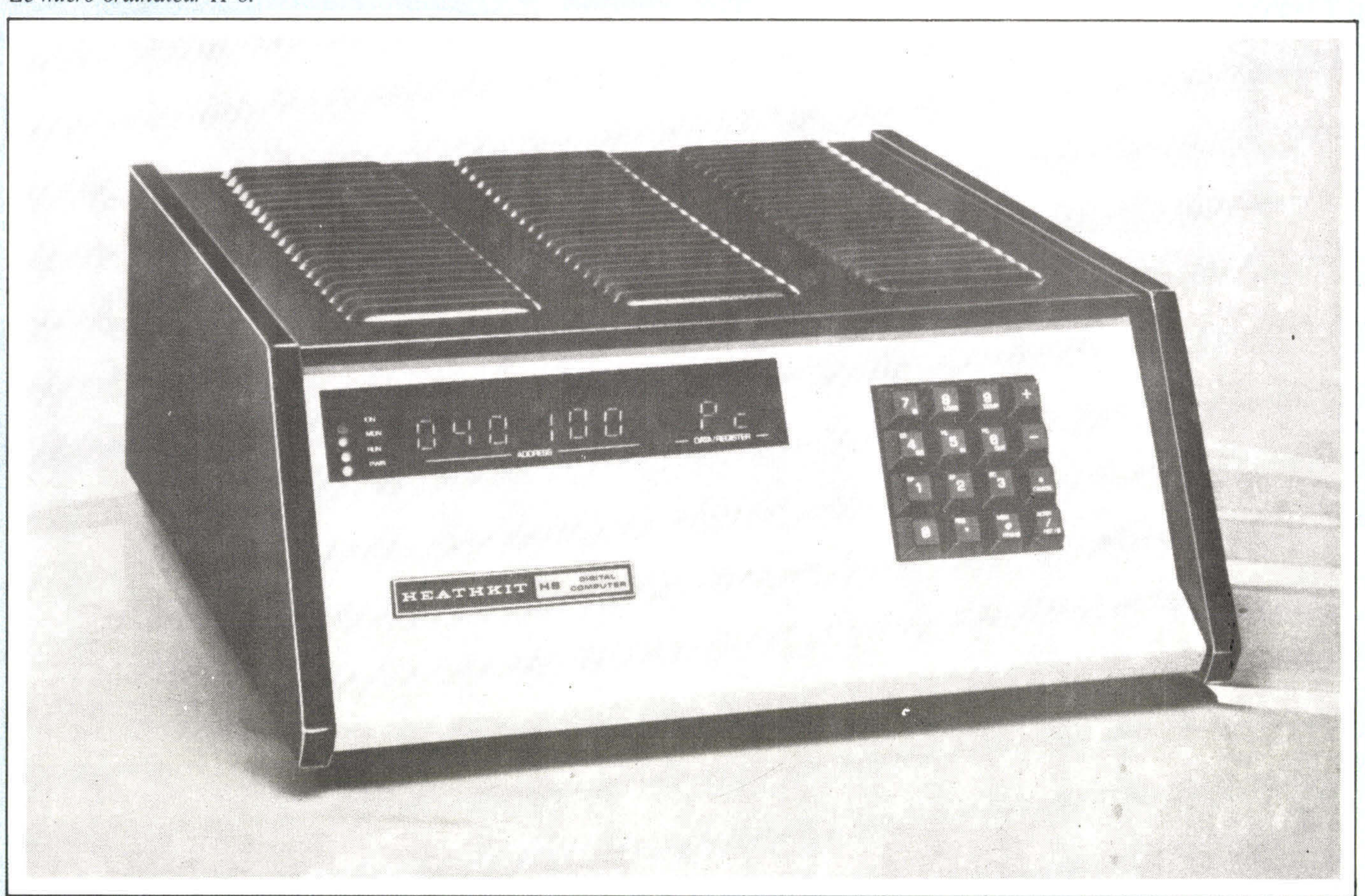
Rappelons que le constructeur propose des systèmes à 8 k RAM pour environ 8000 F en kit et les 16 k RAM pour environ 10 000 F.

Par le prix il est destiné aux amateurs. Par les possibilités il est accessible aux professionnels. Rappelons-nous du texte en plusieurs colonnes sur l'écran (rafraîchi) de la console CRT...

---

* *Heathkit : 47, rue de la Colonie, 75013 Paris. Tél. : 588.25.81.*

Le micro-ordinateur H 8.



Le P.E.T.



Le P.E.T. dans une application graphique...

En 1976-77, Commodore (fabricant de calculettes bien connu) se livrait à une opération de concentration verticale sans précédent. Avec l'achat de MOS Technology (créatrice de la famille de microprocesseurs 6500, qui a actuellement le vent en poupe), et de compagnies spécialisées dans la fabrication de claviers, de magnétophones, de moniteurs TV etc., elle devenait capable de contrôler d'un bout à l'autre, la fabrication de masse d'un microordinateur complet. Peu de temps après, l'annonce du P.E.T. * bouleversait le marché des ordinateurs personnels aux Etats-Unis. En anglais, PET veut dire petit animal familier, ce qui montre bien sa vocation d'ordinateur au foyer.

Qu'y a-t-il dans le P.E.T. ?

Voyons en effet ce qui est proposé et pour quel prix. Pour 6 450 F H.T. (7 585 F TTC), on obtient un microordinateur complet, en un seul morceau (pas de coffrets épars à relier par des câbles) comportant :

— un clavier 73 touches (caractères habituels + caractères graphiques) touches numériques à part,

— un écran de visualisation 23 cm, 25 lignes de 40 caractères avec possibilités graphiques,

— un magnétophone à cassettes pour le stockage de programmes ou de données — Le démarrage et l'arrêt entre enregistrements sont commandés par le P.E.T.,

— une unité centrale à microprocesseur 6502, munie d'une horloge temps réel au 1/60e de seconde,

— 8 k-octets de RAM dont 7 k sont disponibles pour l'utilisateur,

— 14 k octets de ROM comportant notamment l'interpréteur BASIC (8 k) et le moniteur de gestion du système.

Quatre connecteurs relient le P.E.T. au monde extérieur.

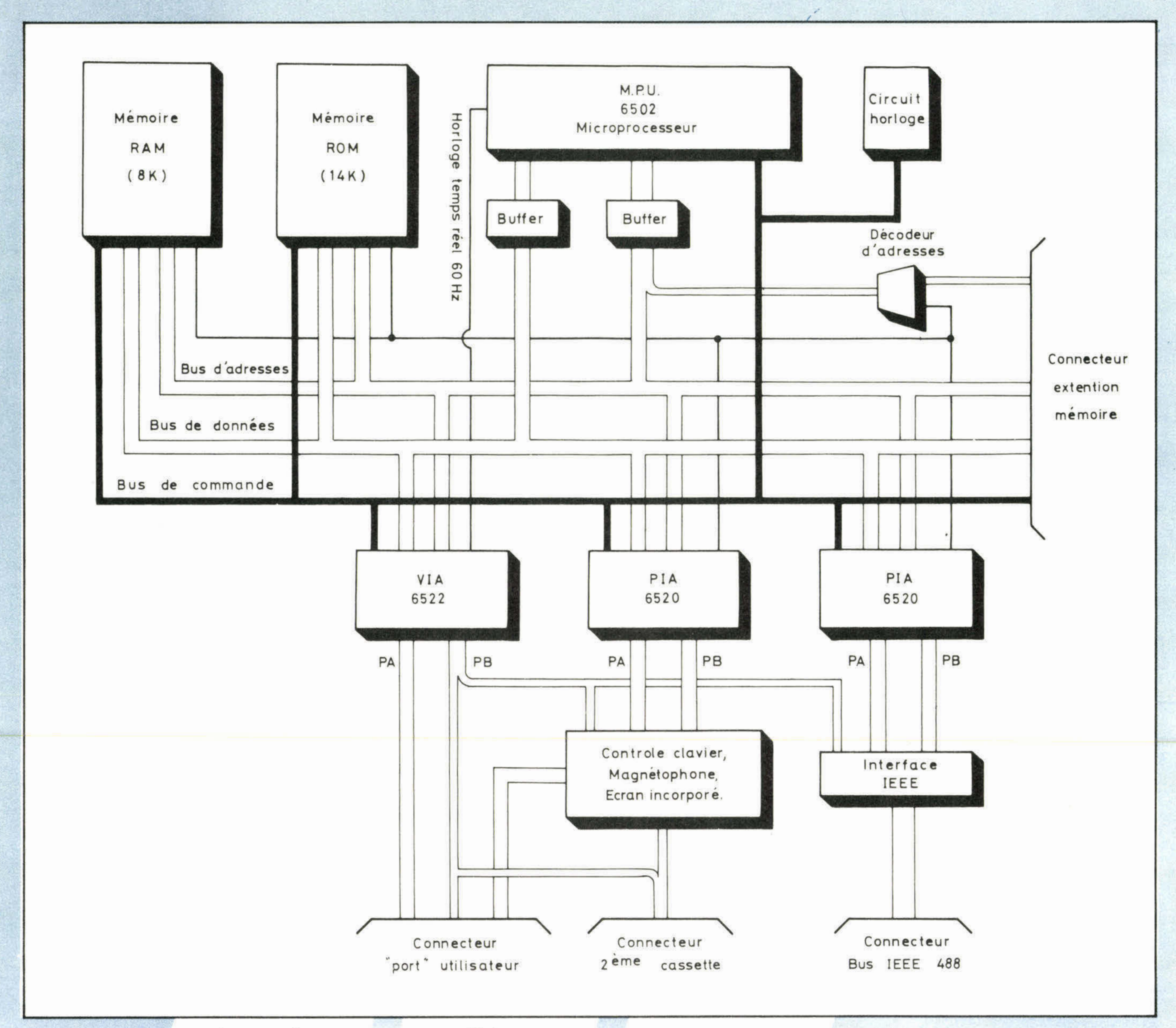
1) Un connecteur 40 contacts transporte les signaux des bus du microprocesseur, autorisant toutes extensions mémoire ou entrées-sorties. On a l'avantage de disposer des signaux de sélection déjà tous décodés par blocs de 4 k.

Les suivants sont plus exceptionnels :

2) Un connecteur pour 2e magnétophone. Le software de base de gestion est déjà présent dans le moniteur. On peut avec le 2e magnétophone aborder toutes les applications de gestion séquentielle de fichiers avec mises à jour.

3) Un connecteur « utilisateur » comportant notamment un port d'entrées-sorties parallèles 8 bits entièrement libre pour l'utilisateur. Allumer une lampe avec le P.E.T.

** Le P.E.T. est distribué par :*
PROCEP, 51, rue de la Rivière,
B.P. 24, 78420 Carrières-sur-Seine. Tél. : 968.31.97.



Synoptique du micro-ordinateur. Le microprocesseur utilisé est un 6502 de MOS Technology.

nécessite uniquement un relais et une instruction dans un programme BASIC.

4) Un connecteur fournit un bus standard IEEE 488 (HP-IB) permettant de relier tout appareil d'instrumentation conforme à ce standard. Il existe des centaines de tels équipements sur le marché, notamment Hewlett-Packard, Tektronix, etc. C'est sur ce bus que se relient l'imprimante 80 colonnes du P.E.T. et les disques qui seront disponibles prochainement.

Le basic du P.E.T.

Le basic du P.E.T. a été développé par Microsoft. Il est analogue à celui du MITS 8800. C'est dire qu'il est certainement un des plus performants du marché sous deux aspects :

a) Extension du langage

Ce BASIC autorise les nombres entiers, les réels (avec 9 chiffres de précision), les variables à dimensions multiples, les chaînes de caractères (fonction LEF$, RIGHT$, MID$, CHR$, ASC, LEN, VAL, STR$). Comme tous les BASIC, celui du P.E.T. ne prend en compte que des noms de variables à 2 caractères au plus, mais il autorise — pour obtenir des identificateurs parlants — des noms plus longs.

Les variables spéciales TI et TI$ permettent de connaître l'heure et de générer des délais grâce à l'horloge temps réel — avantage très rare dans cette catégorie de matériels. La variable ST donne le « status » de la dernière entrée-sortie effectuée.

Le P.E.T. a les fonctions arithmétiques (SIN, TAN, ...) habituelles ainsi qu'un générateur de nombres aléatoires (RND). Notons que le P.E.T. est la seule machine — à notre connaissance — qui « connaisse » la valeur de π (il y a une touche π).

Les opérateurs logiques permettent des traitements au niveau du bit.

Exemple :

Est-ce que le bit 1 de l'adresse 59471 (port extérieur) est à 1 ?

IF PEEK (59741) AND 2 < > 0...

Le P.E.T. a des commandes de manipulations de fichiers très simples mais efficaces, d'autant qu'elles s'appliquent aussi à n'importe quel périphérique du bus IEEE-488 (OPEN, PRINT #, INPUT #, GET #, CLOSE, CMD).

Enfin, les commandes d'accès mémoire PEEK et POKE et d'appel d'un sous-programme en langage machine offrent des possibilités infinies.

b) Rapidité d'exécution

Le basic du P.E.T. est certainement un des plus rapides du marché.

Traitements graphiques

On dispose de possibilités graphiques impressionnantes pour une machine de ce prix. Il y a 64 caractères graphiques du type lignes fines, rectangles, segments circulaires, ronds, croix ainsi que les symboles des cartes à jouer : cœur, carreau, trèfle, pique (voir clavier ci-dessous).

De plus, chacun des caractères peut apparaître en contraste inverse, ce qui fait en tout 256 symboles (lettres) dont 128 graphiques.

La résolution normale est de 25 x 40, mais en jouant sur les symboles employés on peut faire du 200 x 40 ou du 50 x 80.

Enfin, le P.E.T. peut être mis dans un mode spécial, où, au lieu des caractères graphiques, on obtient les lettres minuscules.

Contrôle du curseur

La possibilité de déplacer le curseur dans tous les sens permet de corriger facilement un programme sans retaper toute une instruction, grâce à la touche insertion-suppression (INST).

Exemple :

Si on a tapé par erreur,

10 IF A < > 0 100

(il manque THEN), on amènera le curseur au niveau du 1 de 100. On pressera 4 fois INST puis THEN RETURN et la ligne est corrigée.

Autre exemple :

Supposons qu'on veuille constituer une cascade de tests du genre

10 IF A$ = « A » GO TO 100
20 IF A$ = « B » GO TO 200, etc.

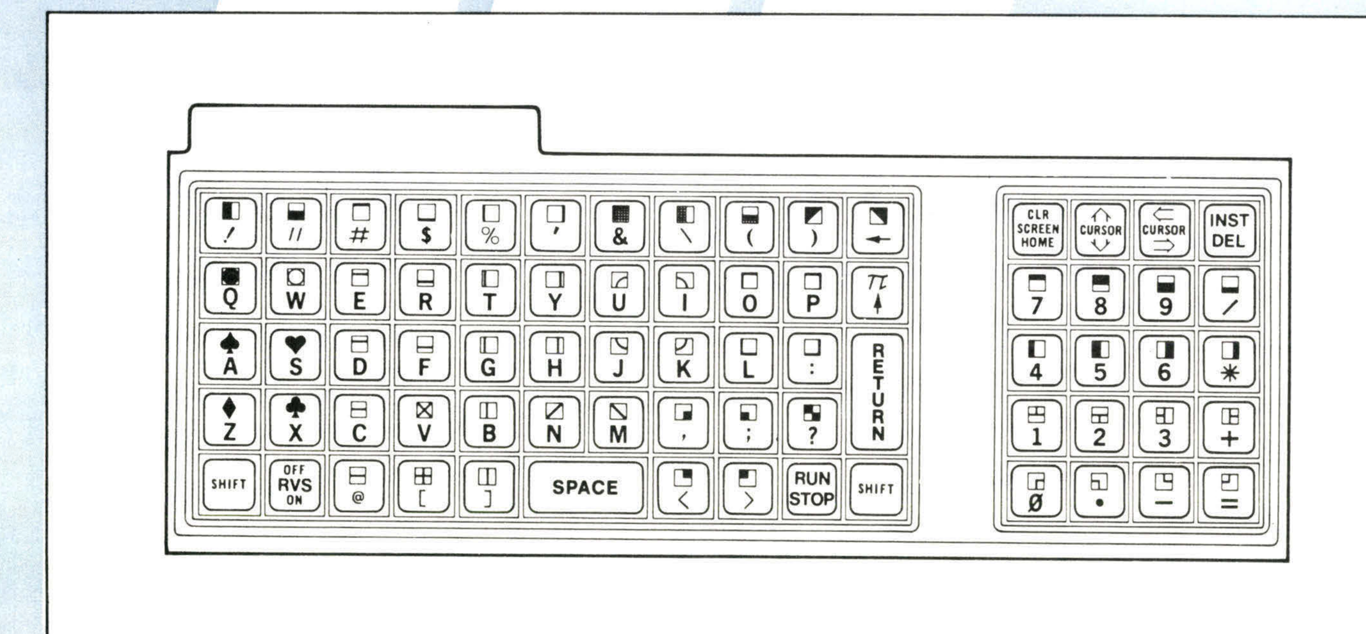
On ne tapera que la première ligne. Ensuite on changera le 1 de 10 et de 100, et le A en B puis RETURN. On aura alors sur l'écran la 2[e] ligne. Mais la première sera restée en mémoire.

En outre, le mouvement du curseur peut être commandé par programme ce qui offre des possibilités intéressantes. Couplé à une commande d'extinction de l'écran (sans vidage), ceci permet de créer d'extraordinaires dessins animés.

Magnétophone à cassettes

Le magnétophone incorporé permet d'abord de sauver un programme sous un certain nom (SAVE), puis de le recharger (LOAD). Noter que le nom utilisé lors du LOAD peut être une abréviation du nom de départ, ce qui est commode. Enfin, s'il s'agit du premier programme présent sur la cassette, son chargement et le lancement de son exécution se font par appui sur une seule touche. Le système indique à l'utilisateur par un message sur l'écran, quelles touches du magnétophone il doit manœuvrer, par exemple, pour une lecture, on doit enfoncer la touche PLAY. En fin de lecture, le magnétophone est arrêté automatiquement par l'ordinateur, et si, pour une lecture ultérieure, on a laissé la touche PLAY appuyée, alors le démarrage est automatique.

Le clavier du P.E.T. comporte 73 touches. Notez la présence des nombreux caractères graphiques ainsi que les symboles des cartes à jouer.



Une commande VERIFY compare le contenu de la mémoire avec ce qui vient d'être écrit sur la cassette : ceci permet de s'assurer qu'une sauvegarde s'est correctement effectuée.

On dispose enfin de commandes de gestion de fichiers de données sur la cassette. Les fichiers de données sont, eux aussi, dotés d'un nom qui permet de les retrouver.

Extensions hardware

Les connecteurs déjà décrits permettent toutes sortes d'extensions. Parmi les extensions prévues, nous pouvons citer (2e magnétophone, imprimante, disques, mémoire (RAM, PROM, REPROM), interface RS232 (U24), boucle de courant, programmateur d'EPROM.

Ici, un point mérite clarification. On peut, bien entendu, remplir les 64 K d'espace adressable du 6502 par de la mémoire du type RAM ou ROM que l'on veut. Alors pourquoi la notice (notons au passage qu'une notice expliquant l'essentiel existe en français) dit-elle qu'on ne peut aller que jusqu'à 32 K de RAM ? Ceci est dû à une particularité d'ailleurs intéressante du BASIC. Lorsqu'on met le système sous tension, on se trouve aussitôt sous BASIC (celui-ci est en ROM : il est indestructible et, en cas de fausse manœuvre, il suffit d'éteindre et de rallumer le P.E.T. pour reprendre le contrôle, sans avoir à recharger le système). Après initialisation, le BASIC explore la mémoire à partir de zéro pour déterminer l'emplacement mémoire disponible. Cette exploration s'arrête à l'adresse 32 K. Ainsi, le BASIC ne tient pas compte de la RAM qui se trouverait au-delà pour implanter les variables et les chaînes de caractères. Mais toutefois, il peut accéder à cette mémoire par des PEEK et des POKE. L'intérêt d'avoir de la RAM au-delà de 32 K est de constituer une zone mémoire « protégée » pour les manipulations en langage machine. A part cela, on peut implanter des ROM ou des REPROM aux adresses souhaitées.

Extensions software

D'innombrables programmes soit de jeux, soit d'applications de gestion sont disponibles dans le cadre du club P.E.T.-France.

Il est aussi possible de se procurer des programmes de base, tels que moniteurs langage machine, assembleurs et désassembleurs. ■

Le BASIC

Nous avons commencé, dans le précédent numéro de Micro-Systèmes, à étudier les premiers rudiments de ce langage en introduisant successivement les notions d'algorithme, d'organigramme et en traitant, directement en BASIC un exemple simple de programme comme celui du calcul du volume d'une sphère $\frac{4}{3}\pi R^3$.

Ce petit programme de quatre lignes n'avait qu'un seul but : vous démontrez que ce langage de programmation est facilement et rapidement assimilable par les non informaticiens.

Vous vous êtes déjà un peu familiarisé avec des instructions telles que INPUT, LET, PRINT et END. Aujourd'hui, nous reprenons rapidement sous une forme un peu différente ces quatre instructions de façon à ce que vous en saisissiez bien la syntaxe et nous avancerons dans l'étude de ce langage en analysant des instructions couramment utilisées en BASIC : READ, DATA, GOTO, IF...THEN, FOR et NEXT, DIM, REM, GO SUB et RETURN, STOP, RUN, NEW, CONT et LIST.

Outre celles énoncées précédemment trois nouvelles instructions spécifiques aux micro-ordinateurs les plus usuels vous sont présentées : PEEK, POKE et USER.

Bien entendu, nous vous proposons un exemple de programme BASIC : aujourd'hui la visualisation d'une courbe sinusoïdale utilisant quelques unes des instructions étudiées dans cet article.

```
LOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVE
L            OVELOVELOVELOVELOVELOVELOV         LOVELOVELOVE
LOV        ELOVELOVELOVELOVELOVELOV                 LOVELOVE
LOVE      VELOVELOVELOVELOVELOVEL                     VELOVE
LOVE      VELOVELOVELOVELOVELOVE            LOVEL      ELOVE
LOVE      VELOVELOVELOVELOVELOV           VELOVELO      LOVE
LOVE      VELOVELOVELOVELOVELOV          OVELOVELOV     LOVE
LOVE      VELOVELOVELOVELOVELOV         LOVELOVELOV     LOVE
LOVE      VELOVELOVELOVELOVELOV        ELOVELOVELO      LOVE
LOVE      VELOVELOVELOVELOVELOV       VELOVELOVEL       LOVE
LOVE      VELOVELOVELOVELOVELOV      OVELOVELOVE        LOVE
LOVE      VELOVELOVELOVELOVEL V     LOVELOVELOV         LOVE
LOVE      VELOVELOVELOVELOVEL V     LOVELOVELO          LOVE
LOVE      VELOVELOVELOVELOVE  V      OVELOVEL           LOVE
LOVE      VELOVELOVELOVELOV   V       VELOV             LOVE
LOVE      VELOVELOVELOVEL     VE                       ELOVE
L                             VELOV                 LOVELOVE
L                             VELOVELOV         LOVELOVELOVE
L             VELOV                                        E
L             VELOV                                        E
LOVE      VELOVELOVELOV   VELOVELOVE      VELOVELOVELO     E
LOVEL      ELOVELOVELO   OVELOVELOVE      VELOVELOVELOVE   E
LOVEL      ELOVELOVELO   OVELOVELOVE      VELOVELOVELOVEL  E
LOVELO      LOVELOVEL   LOVELOVELOVE      VELOVELOVELOVELO E
LOVELO      LOVELOVEL   LOVELOVELOVE      VELOVEL VELOVELOVE
LOVELOV      OVELOVE   ELOVELOVELOVE      VELOVE  VELOVELOVE
LOVELOV      OVELOVE   ELOVELOVELOVE              VELOVELOVE
LOVELOVE      VELOV   VELOVELOVELOVE      VELOVE  VELOVELOVE
LOVELOVE      VELOV   VELOVELOVELOVE      VELOVEL VELOVELOVE
LOVELOVEL      ELO   OVELOVELOVELOVE      VELOVELOVELOVELO E
LOVELOVEL      ELO   OVELOVELOVELOVE      VELOVELOVELOVEL  E
LOVELOVELO      L   LOVELOVELOVELOVE      VELOVELOVELOVE   E
LOVELOVELO          LOVELOVELOVELOVE      VELOVELOVELO     E
LOVELOVELOV        ELOVELOVELOVE                           E
LOVELOVELOV        ELOVELOVELOVE                           E
LOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVE
```

« LOVE »

Dessin sur micro-ordinateur réalisé par la revue américaine « Creative Computing » (Morristown, New Jersey). Ce LOVE est conçu autour d'un programme BASIC très court puisque sa longueur est de 34 lignes.

Problèmes de langage : Basic et langue maternelle

Comme vous pouvez le constater, ce langage pose moins de problèmes aux anglophones, qui voient le programme de calcul du volume de la sphère sous la forme :

50 ENTRER R	(INPUT R)
100 SOIT V= ...	(LET V= ...)
150 IMPRIME « TEXTE » , V	(PRINT ...)
200 FIN	(END)

Il existe dans chaque pays une version « nationale » de Basic. Quelle que soit la langue comprise par la machine, il faut pour programmer :

— être concis, en respectant les quelques mots que la machine connaît et leur enchaînement en propositions simples,

— ne pas demander des choses impossibles ou peu claires. La machine ne fait qu'interpréter et exécuter les ordres. L'art de programmer est aussi un art de commander.

Imaginons dans l'exemple précédent que l'instruction 200 (elles sont exécutées dans l'ordre croissant des numéros qu'on leur attribue) soit remplacée par les deux instructions suivantes :

```
200 VA EN 50
250 FIN
```

C'est l'exemple d'un programme « qui n'a pas de sens », une fois lancé il produirait l'impression d'une suite de lignes :

LE VOLUME DE LA SPHÈRE EST DE 7.8819...

LE VOLUME DE LA SPHÈRE EST DE 7.8819...

ETC.

Car, après l'exécution de l'instruction d'impression 150, le programme rencontre l'instruction

200 qui lui demande de recommencer.

A l'image de l'apprenti sorcier, nous ferons tourner notre « balai » à l'infini. Il remplit des feuilles et des feuilles avec le même message dont il calcule les valeurs en quelques millionièmes de seconde. On dit que le programme **« boucle »** ou **« se plante ».**

Comment l'arrêter ? En appuyant sur une touche « STOP » ou « Remise à zéro » s'il y en a une ou en coupant purement et simplement le courant...

Dans les lignes que nous avons placées en 200, il apparaît un nouvel ordre : « GO TO » (VA EN), de saut. La programmation en Basic s'apprend en quelques heures et s'exprime par une suite d'instructions assez claires. Le tout est de bien les ordonner les unes à la suite des autres, sans demander des choses impossibles du genre :

```
50 LET X = 3 X³ - 2 X² + 251
100 PRINT X
```

La machine ne pourra pas résoudre toute seule l'équation cubique, l'instruction LET (SOIT) n'étant qu'une simple affectation de valeur à X. Si, par exemple, l'instruction précédente donnait à X une valeur quelconque,

```
25 X = 5
```

Dans le meilleur des cas on obtiendrait l'impression d'un résultat correspondant au calcul de

$$X = 3.5^3 - 2.5^2 + 251 = 576$$

et, pour la suite du programme, X aura cette nouvelle valeur qui « écrasera » celle qu'il avait initialement.

Précisons que le signe égal « = » n'a pas le même sens en informatique qu'en mathématiques. Pour mieux le souligner, les « esthètes » de la programmation le remplacent par « : = » qui est par convention l'opérateur d'affectation. Donc ne nous étonnons pas de voir des instructions du genre :

```
50 X = X + 1
```

Elles signifient en pratique : calcul de l'expression à droite du signe égal et affectation du résultat à la variable qui se trouve à gauche, même si les opérations s'effectuent sur une variable identique.

Pour résumer disons que tout programme doit répondre à deux conditions :

— Il doit d'abord être écrit dans le langage compris par le micro-ordinateur. Ce langage est composé de symboles mathématiques et de mots en anglais ou en français si la langue de la machine est le français. La signification des symboles mathématiques peut présenter des particularités elle s'apprend sur une table d'opérateurs.

— La seconde de ces conditions est que tout programme doit être établi de façon complète et précise. La machine n'est pas intelligente et ne peut pas deviner ce qu'on a l'intention de lui dire. Reprenons donc systématiquement.

Rédaction d'un programme

Seules les lettres majuscules sont utilisées, les lignes commencent par un **numéro de ligne** qui sert à identifier chaque **instruction.** Les numéros de ligne servent à indiquer dans quel ordre la machine doit exécuter les instructions. On peut enregistrer un programme dans n'importe quel ordre. On peut insérer des instructions en leur donnant un numéro d'ordre intermédiaire, même après l'écriture complète d'un programme. Par exemple « 25 » pour placer une nouvelle instruction entre « 20 » et « 30 ». Avant le traitement le micro-ordinateur procède à un classement dans l'ordre indiqué par les numéros de ligne des instructions à exécuter.

A la suite du numéro de ligne, chaque instruction débute par un mot ou une expression.

Une ligne sans numéro est exécutée immédiatement après le retour-chariot.

Les espaces n'ont aucune signification en langage Basic, excepté dans les messages qui doivent être imprimés. Ils peuvent servir à rendre plus facile la lecture d'un programme :

```
10 . . . . LET . . X = 5
10LETX=5
```

ont la même signification.

Les formules doivent tenir sur une seule ligne. Elles utilisent cinq opérations arithmétiques : l'addition, la soustraction « + », « - », la multiplication « * » (astérisque), la division « / » (le slash) et l'élévation à une puissance « ↑ ».

L'utilisation des parenthèses « (», «) » est la même qu'en mathématiques. La machine effectue d'abord le calcul indiqué entre parenthèses, le résultat ainsi obtenu sert ensuite aux autres calculs.

Le langage permet l'utilisation des fonctions mathématiques suivantes :

```
SIN (X) ; COS (X) ; TAN (X)
                    ATN (X)
```

signifiant le calcul du sinus, cosinus, de la tangente et de l'arc tangente de X, ce dernier étant soit un nombre, soit un angle mesuré en radians.

```
EXP (X) ; LOG (X) ; SQR (X)
```

l'exponentielle de X (e^X), logarithme naturel de X (ln X) et la racine carrée ($\sqrt{X}$) ; ainsi que

```
ABS (X), valeur absolue |X|
INT (X), valeur entière (INT
                  (3,14) = 3)
```

Les NOMBRES peuvent être positifs ou négatifs, exprimés sous forme décimale avec un maximum de 9 chiffres. Exemples : 2, - 3.675, 123456789, -.123456789, 1234.56789.

La notation polonaise est également acceptée. La lettre « E » signifie dans ce cas « multiplication avec 10 à la puissance indiquée ».

Ainsi :

```
1978 = 197 . 8 E1
     = 1 . 978 E3
     = 197800 E-2
```

Toutes ces notations sont acceptées.

On ne peut pas écrire E2, car cela pourrait passer pour une variable. Les puissances de 10 seront notées sous la forme 1E2, pour préciser qu'il s'agit de multiplier 1 par 10^2.

Les VARIABLES en Basic sont représentées par une seule lettre ou par une lettre suivie d'un seul chiffre : A, X, N5, E2. Une variable correspond à un nombre qui n'est pas connu par le programmeur au moment où il rédige le programme.

Les valeurs sont assignées aux variables par les instructions LET ou READ.

Les SYMBOLES DE RELATION. Ils sont utilisés dans les instructions IF... THEN dans lesquelles il y a comparaison des valeurs :

= égal à (égal mathématique n'est pas une affectation)
< inférieur
<= égal ou inférieur
> supérieur
>= supérieur ou égal
<> différent de

Le jeu d'instructions

Le nombre d'instructions mis à la disposition de l'utilisateur peut varier d'une machine à l'autre. Néanmoins, nous pouvons faire appel au tronc commun suivant :

PRINT

PRINT 3.08 ↑ 2.54, cette instruction vous permet d'obtenir le résultat d'un calcul compliqué dès la frappe de la touche « Retour Chariot » (CR).

Autres formes de PRINT

PRINT seul sert à sauter une ligne.

PRINT « message à paraître tel quel » ; variable 1 ; variable 2... on obtient le message suivi des valeurs des deux variables.

```
50 PRINT USING 100 « Nom
   du fournisseur », N, I
100 : « ... » ### . ## ##
```

La ligne 100 dans cet exemple est une ligne-image, donnant le format d'impression.

Un « ; » à la fin d'un PRINT fait suivre l'impression d'un autre PRINT sur la même ligne.

LET

Cette instruction n'indique pas une égalité algébrique. Elle ordonne à la machine d'effectuer les opérations demandées et d'assigner le résultat à une variable donnée.

Exemple. Calculons ce que devient un capital C, investi à T % par an en N années.

L'équation est de la forme $X = C\,(1 + T/100)^N$, c'est-à-dire en Basic :

```
 50 LET X = C*(1 + T/100)↑N
100 PRINT  « ANNEES : », N, "CAPITAL = ", X
 40 T = 12
 30 N = 15
 20 C = 150000
110 END.
```

GO TO (numéro de ligne)

Cette instruction renvoie la machine à la ligne désirée. Ouvre la porte aux programmes qui « bouclent » (tournent sans fin).

10 GO TO 10 représente une boucle sans fin.

Il existe aussi un :

```
GO TO n1, n2, ... nm ON
(expression).
```

Il y a saut à l'adresse n_i si la partie entière de l'expression calculée est égale à i.

IF THEN

On peut avoir à sauter la suite normale des instructions lorsqu'on rencontre une relation donnée. La forme la plus générale de cette instruction est :

IF (expression arithmétique) (relation) (expression arithmétique) THEN (numéro de ligne ou instruction)

READ et DATA

L'instruction READ sert à assigner aux variables qui sont indiquées les valeurs figurant dans une instruction DATA. Les READ figurent généralement vers le début du programme, les DATA sont situées à une place arbitraire, habituellement avant l'instruction END.

Exemple :

```
100 READ A, B, C, D, F, G
200 DATA 5, 10, 15
300 DATA 20, - 123.456 E -5, 25
```

123.456 E – 5 correspond à la variable F, car les données sont attribuées dans l'ordre d'apparition dans READ.

Exemples

```
20 IF ABS (X) < = N THEN 40
40 IF N > = ABS (X) THEN 20
```

La première de ces instructions recherche si la valeur absolue de X est inférieure ou égale à N et si c'est le cas, donne l'ordre de sauter à la ligne 40.

La seconde instruction recherche si N est supérieur ou égal à ABS (X), auquel cas la machine doit sauter à la ligne 20.

Si ces deux instructions se trouvaient dans un programme, on tourne indéfiniment si $N \geqslant ABS(X)$.

FOR et NEXT

Ces deux instructions définissent le début et les conditions

(FOR) puis la fin (NEXT) d'une boucle. Le cliché de l'instruction FOR est le suivant :

FOR (variable) = (expression arithmétique) TO (expression arithmétique) STEP (expression arithmétique)
...
...
...
NEXT (variable)

Voir exemples ci-contre.

Les bornes de variation de la variable peuvent être positives, négatives, ou à calculer, la raison de la progression(ou le pas : step) pouvant être du même ordre.

Il n'y a aucune commune mesure avec une boucle DO... CONTINUE en FORTRAN.

Si la première et la dernière valeur ainsi que l'écart sont donnés sous forme d'expressions arithmétiques, celles-ci sont calculées une fois pour toutes lors du premier passage par l'instruction FOR. La variable qui suit FOR peut être modifiée au cours de la boucle ; le test de sortie (vérification de la dernière valeur une fois l'écart ajouté) porte toujours sur la valeur finale de cette variable.

Exemples :

```
1) 30 FOR X = O TO 3 STEP D
   ,
   ,
   ,
   80 NEXT X

2) 50 FOR I = (1 + SQR (Z)/3) TO 3 * 1E5↑1.35 STEP 1/4
   ,
   ,
   ,
   100 NEXT I

3) 10 FOR A = 8 TO - 5 STEP - 1
   ,
   ,
   ,
   20 NEXT A
```

DIM

Il convient d'utiliser cette instruction pour l'introduction des tableaux. La machine prévoit dans ce cas l'espace nécessaire :

```
20 DIM A (35)
30 DIM B (5,25)
```

La première instruction permet d'introduire une liste de 36 éléments, par la méthode suivante :

```
10 DIM A (35)
20 FOR I = O TO 35 STEP 1
30 READ A(I)
40 NEXT I
,
,
,
90 DATA 1, 2, 3, 4, 5, 6, 7...34,
   35, 36
100 END
```

L'indice « 0 » est permis, contrairement à ce qui est autorisé dans d'autres langages.

La seconde instruction permet d'introduire une table de 6 x 26 éléments par une méthode rapprochante. Les tableaux peuvent avoir plusieurs dimensions. Les réservations se font dans la limite des disponibilités de mémoire : DIM A (1000, 1000) ne peut être conçu que sur une machine disposant au moins d'une capacité de 1000 X 1000 mots au moment où l'instruction DIM est rencontrée.

REM

Cette instruction sert à l'introduction de lignes de commentaire dans le texte (remarques). Ces lignes ne sont pas exécutées par le programme même si elles contiennent des mots-Basic et rendent intelligibles les programmes.

INPUT

Par cette instruction on peut entrer des données accompagnées d'un texte de dialogue, tel que :

```
10 INPUT
      « VOTRE  AGE ? », A
```

fait apparaître au moment de l'exécution :

```
VOTRE AGE ? 19
```

La machine s'arrête en attente de la variable A. Elle repart après le retour-chariot. S'il n'y a pas de texte à imprimer après le mot INPUT, le signe d'interrogation apparaît à l'exécution en début de ligne.

Exemple :

```
10 INPUT A, B, C, D
20 X = (A + B + C + D)/4
30 REM  CE  PROGRAMME CALCULE LA
40 REM  MOYENNE  ARITHMETIQUE
50 PRINT X
60 END

RUN
?  10, 20, 30, 40 CR
25
```

Un message d'erreur apparaît si le nombre de variables entrées ne correspond pas au nombre demandé.

GO SUB et RETURN

Ces instructions permettent l'exécution d'une portion d'un programme à plus d'une reprise. GO SUB indique le numéro de ligne de la première instruction du sous-programme.

```
 90 GOSUB 200
100 . . .
 ,
 ,
 ,
200 X = (A + B + C + D)/4
210 PRINT X
220 RETURN
```

Le sous-programme est placé à la ligne 200 et calcule la moyenne arithmétique de quatre nombres.

L'instruction RETURN renvoie la machine à la première ligne dont le numéro est supérieur à 90, ici 100.

On peut utiliser l'instruction GO SUB à l'intérieur d'un sous-programme pour exécuter un autre sous-programme, on obtient alors un « GO SUB emboîté ». Le retour d'un sous-programme par un GO TO ou un IF-THEN n'est pas autorisé.

STOP

Cette instruction correspond à l'instruction GO TO N dans laquelle N est le numéro de ligne de l'instruction END du programme. Voici deux programmes équivalents :

```
 50 GO TO 100      50 STOP
 ,
 ,
 60 GO TO 100      60 STOP
 ,
 ,
100 END           100 END
```

END

Cette instruction annonce la fin du programme.

Mis à part les instructions, l'on dispose de plusieurs DIRECTIVES :

RUN

Permet le lancement d'un programme à partir du plus petit numéro de ligne existant. Une ligne sans numéro est exécutée immédiatement. Ainsi, un GO TO sans numéro permet, par son exécution immédiate, le même lancement de programme.

```
 10 LET . . .
 ,
 ,
 ,
100 END
GO TO 10
```

NEW

Cette directive efface toute trace de programme, afin de pouvoir écrire un nouveau programme-utilisateur.

CONT

Elle est utilisée lorsque l'on rencontre un STOP si on veut CONTinuer.

LIST

Elle permet d'obtenir une liste du programme entrée en mémoire.

Particularités du Basic implanté sur les micro-ordinateurs actuels

Pour les utilisateurs actuels du Basic les constructeurs ont prévu des instructions pouvant manipuler des mots-mémoire dont on spécifie l'adresse en binaire-hexadécimal exprimé en décimal ou permettant le lancement de programmes ou de sous-programmes en langage machine, donc d'une exécution très rapide, à partir du texte de commandes en Basic.

Ces instructions, rencontrées sur le PET, PROTEUS, APPLE, etc., sont le **PEEK, POKE** et **USER.**

PEEK*

Cette fonction est utilisée pour lire ou saisir le contenu d'un emplacement mémoire dont on spécifie l'adresse sous forme **décimale.**

Exemple :

```
LET A = PEEK (10000).
```

Affecte à la variable A la valeur de la case mémoire d'adresse 10000 en décimal, ou 2710_H en hexadécimal.

POKE*

Elle permet d'écrire dans un mot mémoire. Sa structure : POKE (Adresse en décimal, Valeur en décimal).

Exemple :

```
100 POKE (10000, 20).
```

Cette instruction écrira $20_{10} = 14_H = 00010100$ dans la case mémoire d'adresse 10000 (2710 Hexa).

USER*

Elle permet d'appeler un sous-programme écrit par l'utilisateur en langage machine, sous contrôle du Basic. L'écriture, la lecture, la mise au point et contrôle du programme-machine se font à partir du PEEK et POKE.

L'instruction USER exécutera un sous-programme dont l'adresse de début est contenue dans des mots mémoire bien définis par le constructeur de la machine.

Telle est donc la REDACTION d'un programme en Basic. Nous faisons maintenant place aux applications. Aujourd'hui, le programme de visualisation d'une courbe.

---

* *PEEK : regarder (à la dérobée).*
* *POKE : pousser.*
* *USER : usagé (au sens utilisateur).*

Programme de visualisation de courbes ou d'histogrammes *

Ce programme de quelques lignes permet de visualiser sous forme de points ou d'astérisques une fonction ou une expression quelconque, Z.

En l'occurrence, dans cet exemple, nous traiterons le cas d'une sinusoïde d'équation : Z = SIN (X).

Analysons chacune des instructions que comporte ce programme.

Ligne 10
La ligne 10 est le début d'une boucle FOR, dans laquelle X varie de 0 à 2 π au pas de 0,3 radian.

Ligne 15
On calcule la nouvelle valeur du sinus.

Ligne 17
On imprime la valeur numérique et l'on se positionne à 12 espaces du début de ligne pour la tabulation (TAB (12)*).
Remarquer le « ; » qui fera suivre le prochain PRINT sur la même ligne.

Ligne 20
Le résultat Z est compris entre - 1 et + 1. On procède alors à une mise à l'échelle en vue d'obtenir un nombre de points imprimés proportionnel à l'amplitude de la fonction. Le facteur d'échelle est « 15 ». Nous décalons également de « 16 » la fonction, vers la droite, pour pouvoir obtenir les variations positives et négatives sur la ligne.

Ligne 30
Le résultat du calcul de la ligne Y contient des décimales. Nous conservons uniquement la partie entière malgré la perte de précision (elle est au maximum de 1/15).

Ligne 40
L'entier « K », introduit dans une bande d'impression, produit K astérisques, la longueur de la ligne imprimée est ainsi proportionnelle à l'amplitude de Z.

Ligne 70
Cette instruction n'a d'autre but que le retour à la ligne.

Ligne 90
Retour à la ligne 10 (bouclage). Le programme imprimera une sinusoïde sans fin. C'est utile sur des stands de démonstrations ou en vitrine. L'arrêt s'obtient par un CTRL C, Reset, etc. ■

Histogramme : graphique constitué par des rectangles de même base, placés côte à côte, et dont la hauteur est proportionnelle à la quantité à représenter. (Petit Larousse)

TAB (12). En général TAB (n) fait avancer la tête de frappe de n blancs à partir du début de la ligne.

```
#LIST

0010 FOR X=0 TO 2*6.28 STEP .3
0015  Z=SIN(X)
0017 PRINT Z;TAB(12);
0020  Y=15*Z+16
0030  K=INT(Y)
0040 FOR J=1 TO K STEP 1
0050 PRINT ".";
0060 NEXT J
0070 PRINT
0080 NEXT X
0090 GOTO 10
0100 END

READY
#RUN
0           ................
0.2955202   ....................
0.5646425   ........................
0.7833269   ...........................
0.9320391   .............................
0.997495    ..............................
0.9738476   ..............................
0.8632094   ............................
0.6754636   ..........................
0.4273821   ......................
0.1411306   ..................
-0.1577555  .............
-0.4425225  .........
-0.6877665  .....
-0.8715758  ..
-0.9775301  .
-0.9961646  .
-0.9258147  ..
-0.7727645  ....
-0.5506855  .......
-0.2794155  ...........
0.01681391  ................
0.3115414   ....................
0.5784398   ........................
0.7936679   ...........................
0.938       ..............................
0.9985433   ..............................
0.9698898   ..............................
0.854599    ............................
0.6629696   .........................
0.4121209   ......................
0.124466    .................
-0.1743358  .............
-0.4575377  .........
-0.699875   .....
-0.8796958  ..
-0.9809362  .
-0.9945526  .
-0.9193285  ..
-0.7619836  ....
-0.5365729  .......
-0.2632318  ............
0           ................
0.2955202   ....................
0.5646425   ........................
```

Le cheminement des informations dans un MICRO-ORDINATEUR

Microphotographie d'une pastille de microprocesseur, 4 bits rapide SF.C 92901 technologie bipolaire (cliché Thomson-CSF).

Les informations nécessaires au fonctionnement de l'Unité Arithmétique et Logique ou ALU sont stockées dans une mémoire. Cette mémoire contient deux types d'informations : les instructions du programme que l'ALU devra exécuter (addition, soustraction, décalage, ET, OU...) et les données sur lesquelles l'ALU effectuera les traitements dictés par les instructions.

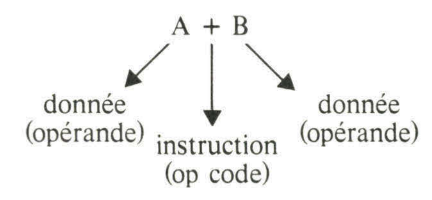
Pour bien comprendre le fonctionnement d'un microprocesseur, il est important de saisir la façon dont les informations transitent et circulent dans le système mémoires/microprocesseur et coupleur d'entrée-sortie.

A titre d'exemple, nous analyserons le cheminement des informations dans un tel système et nous étudierons les différents signaux générés par le séquenceur pour trois types d'instructions classiques : LDA (charger), STA (ranger) et JMP (saut) d'un microprocesseur standard dont le bus de données est de 8 bits et le bus d'adresse de 16 bits.

La transmission séquentielle

Les instructions du programme (informations traitantes) que l'ALU exécute sont aussi appelées : code opération ou **op code.** Les données ou informations traitées portent le nom d'**opérandes.**

Par exemple, dans une opération d'addition les deux nombres A et B sont les opérandes et le signe + le code opération :



Or, à ces deux types d'informations correspondent deux unités particulières. Les données, appliquées à l'Unité Arithmétique et Logique, seront traitées par elle et les instructions seront analysées par l'unité de commande (ou de contrôle).

Le schéma synoptique de la **figure 1** représente le cheminement des instructions et des données. A titre indicatif, le schéma bloc de la **figure 2** montre la constitution d'un microprocesseur désormais bien connu : le 8080 de Intel.

Pour diriger de façon **séquentielle** les informations, soit à l'unité de commande, soit à l'ALU, il faut connaître successivement les **différentes adresses de la mémoire.**

Ces adresses seront obtenues à l'aide d'un compteur, dont la valeur sera augmentée de 1 (on dira incrémentée) après chaque

Un bus est matérialisé par autant de fils en parallèles que de bits à transmettre.

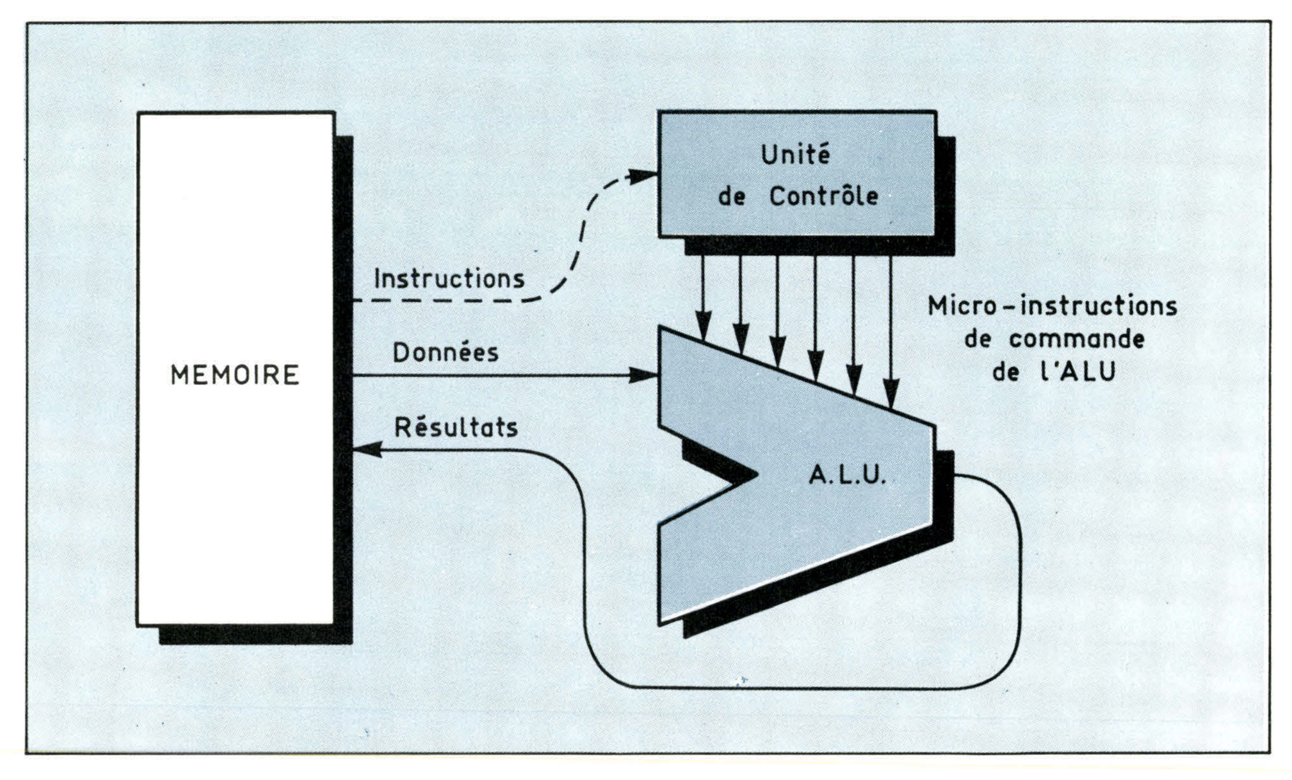


Fig. 1. - Synoptique d'un système à microprocesseurs. Les données seront traitées par l'ALU et les instructions par l'unité de contrôle.

accès à la mémoire. Ce compteur s'appelle le **compteur ordinal** (ou program counter PC).

D'autre part, lorsque les informations arrivent de la mémoire, il est nécessaire de savoir si l'on est en présence d'une instruction ou d'une donnée. Pour résoudre ce problème, on a pris le principe suivant :

- **La première information est toujours une instruction.**
- **La ou les informations suivantes sont des données ou des adresses mémoire.**

Par conséquent, la première information est stockée dans un registre spécial appelé **registre d'instruction.** La sortie de ce registre est décodée. Cette information décodée indique si les octets suivants sont des données, et combien d'octets sont transmis pour cette donnée. Après avoir traité ces données, le registre d'instruction est remis à zéro, et est prêt à recevoir l'instruction suivante.

Les ordres de commande des différents cheminements des informations, sont fournis par l'unité de commande, à partir de la sortie décodée du registre d'instruction. Il existe sur le schéma synoptique de la **figure 3** trois chemins de circulation des informations :

- Le **DATA BUS** (ou bus de données) : sur cette voie circulent les informations allant ou venant à la mémoire (données, instructions).
- Le **BUS ADRESSE :** sur cette voie circulent les informations fournissant l'adresse des différents points mémoire.
- Le **BUS DE CONTROLE :** sur cette voie circulent les micro-instructions provenant du séquenceur.

Précisons qu'un bus est matérialisé par autant de fils en parallèles que de bits à transmettre. Par exemple, dans un bus de données de 8 bits, il y a 8 fils : D_0, D_1, D_2... D_7.

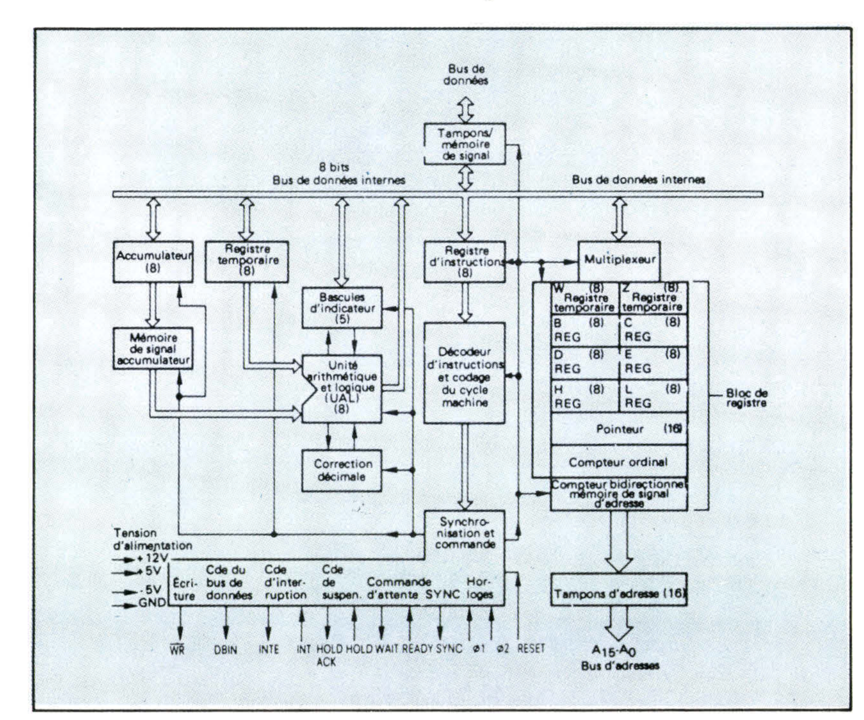


Fig. 2. - Constitution interne du microprocesseur 8080.

Fig. 3. – Synoptique d'un système de traitement de l'information. Remarquez les 3 chemins de circulations : le BUS de données, le BUS d'adresse et le BUS de contrôle.

Fig. 4. – Rôle de l'unité de commande : aller chercher, décoder et exécuter l'instruction.

Principe d'exécution d'une instruction

Toujours dans le cas où le système possède 8 bits de données et 16 bits d'adresses, examinons le principe d'exécution d'une instruction.

Au début de chaque cycle le contenu du compteur ordinal est déposé sur le bus adresse. Les bits du registre du compteur ordinal (tampon) sont dirigés vers la mémoire. Une impulsion de lecture ou d'écriture (read/write) est alors présentée à la mémoire par l'intermédiaire du bus de commande. Si nous avons une lecture par exemple, au bout de 300 à 500 nS (suivant le type de mémoire employée) l'information est disponible et le contenu du mot adressé par le compteur ordinal est placé sur le bus des données (DATA BUS).

Les bits venant de la mémoire sont expédiés dans le registre d'instruction. Cette première phase s'appelle **fetch** (aller chercher).

L'instruction contenue dans le registre d'instruction est décodée. Cette seconde phase s'appelle **decode** (décodage).

Enfin l'instruction décodée est exécutée séquentiellement dans le temps à l'aide du séquenceur (ou unité de commande). Cette troisième phase s'appelle **execute** (exécution).

Si l'instruction nécessite de nombreux bits pour mener à bien son exécution, l'unité de commande donne l'ordre d'aller rechercher en mémoire les bits suivants. Elle doit effectuer la totalité des séquencements et son rôle peut être résumé par le graphe de la **figure 4.**

Chaque fois que l'on accède à la

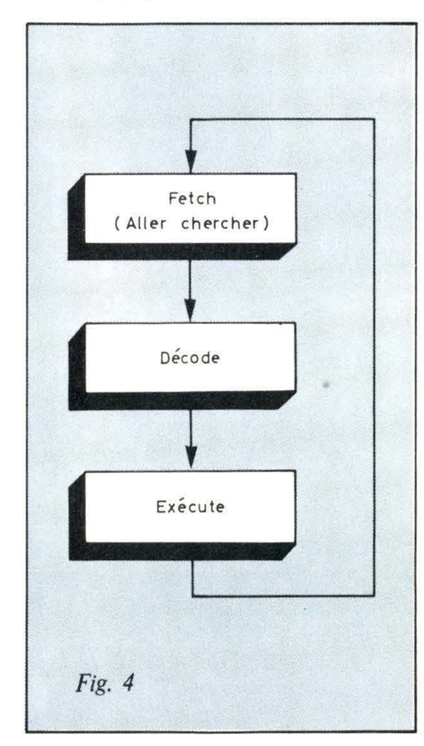


Fig. 4

Pour assurer à la mémoire que l'on envoie une information d'adresse le bus de contrôle lui transmet un VMA (validation adresse mémoire).

mémoire (un fetch), on a par définition un **cycle machine.** Chaque cycle machine requiert de 2 à 5 états internes. Chaque état interne correspond à une **micro-instruction.** Dans la majorité des microprocesseurs, ceci correspond également à une période d'horloge ; les micro-instructions étant générées au rythme du signal d'horloge. On dit que l'on a un **fonctionnement synchrone.**

Description de quelques instructions

Afin de bien connaître le cheminement des informations et pour étudier les signaux générés par le séquenceur, nous allons analyser trois instructions particulières.

- Instruction de chargement :

LDA (adresse)

Charger le contenu de la mémoire désignée par l'adresse dans l'accumulateur (A). LDA est le mnémonique (ou abréviation anglo-saxonne) de LOAD A (charger l'accumulateur).

- Instruction de rangement :

STA (adresse)

Ranger le contenu de l'accumulateur (A) à l'adresse désignée par l'instruction. STA est le mnémonique de STORE A (ranger).

- Instruction de saut :

JMP (adresse)

Lorsque le programme exécute cette instruction il se branche à l'adresse spécifiée et continue le traitement des informations. JMP est le mnémonique de JUMP (sauter).

Etude de LDA (adresse)

Examinons, maintenant, comment s'organisent les différents cycles machine constituant l'instruction LDA.

1er cycle machine : *recherche et décodage de l'instruction*

A la première impulsion d'horloge, le contenu du compteur ordinal est déposé sur la ligne adresse (bus d'adresses). La sortie du registre tampon du compteur ordinal est validée grâce à une impulsion PCO (Program Counter Out).

D'autre part, pour assurer à la mémoire que l'on envoie bien une information d'adresse, on lui transmet par l'intermédiaire du bus de contrôle une information VMA (Valid Memory Adress).

A la deuxième impulsion d'horloge, un ordre de lecture est envoyé à la mémoire par le bus de commande (read). Pendant le temps nécessaire à la recherche de l'adresse à l'intérieur de la mémoire, le contenu du compteur de programme est incrémenté. Cette opération est représentée par la relation :

$$(PC) \longrightarrow (PC + 1)$$

A la troisième impulsion d'horloge, le contenu du mot mémoire est acheminé dans le registre instruction (IR) par l'intermédiaire du bus de données. Pour cela, l'information présente sur le DATA BUS est chargée dans le registre IR par une impulsion IRL (Instruction Register Load).

A la quatrième impulsion, le contenu du registre d'instruction sera décodé. Pour cela on applique une impulsion de code, au décodeur.

Nous pouvons tracer le chronogramme (Timing) des différentes micro-instructions émises pendant ce premier cycle machine, ainsi que le cheminement des informations **(fig. 5).**

La configuration binaire de l'instruction reçue et décodée indique qu'il faut aller chercher en mémoire les deux mots suivant l'instruction. Ces deux mots seront utilisés, comme une adresse qui devra être transférée sur le bus des adresses de manière à être transmise à la mémoire.

2e cycle machine : *recherche de la partie basse de l'adresse*

Le contenu du compteur ordinal est à nouveau placé sur le bus des adresses et l'on recommence le même cycle que précédemment. Mais au bout du temps T_3 l'information présente sur le DATA BUS est envoyée dans le registre d'adresse temporaire à l'aide d'une impulsion ADRL (ADress Register Load).

Nous pouvons tracer le chronogramme des micro-instructions et le cheminement des informations pendant ce deuxième cycle machine **(fig. 6).**

3e cycle machine : *recherche de la partie haute de l'adresse*

Pendant le troisième cycle machine, on va rechercher l'autre partie de l'adresse. La séquence est identique à la précédente, mais l'octet est placé dans ce cas dans la partie haute du registre d'adresse temporaire **(fig. 7).**

4e cycle machine : *chargement de la donnée dans l'accumulateur*

A la première impulsion d'horloge de ce quatrième cycle machine, le contenu du registre adresse temporaire est envoyé sur le bus adresse à l'aide d'une impulsion ADRO (ADdress Register Out). On envoie également une information de validation VMA. La durée de la deuxième impulsion d'horloge est nécessaire à la mémoire, pour rechercher et rendre disponible l'information.

A la troisième impulsion, l'information contenue dans le bus de données est transférée dans l'accumulateur A par une impulsion ALD (Accumulator LoaD). En même temps une impulsion de remise à zéro est envoyée au registre d'instruction CIR (Clear Instruction Register), et le système est prêt pour recevoir l'instruction suivante.

Traçons le chronogramme des micro-instructions et le cheminement des informations de ce quatrième cycle machine **(fig. 8).** Nous pouvons ainsi résumer l'ensemble des micro-instructions émises par le séquenceur pour exécuter l'instruction LDA (Adresse) **(fig. 9).**

Fig. 5. – Chronogramme et cheminement des informations lors du 1er cycle machine de l'instruction LDA. Ce cycle correspond à la recherche en mémoire et au traitement de ces informations.

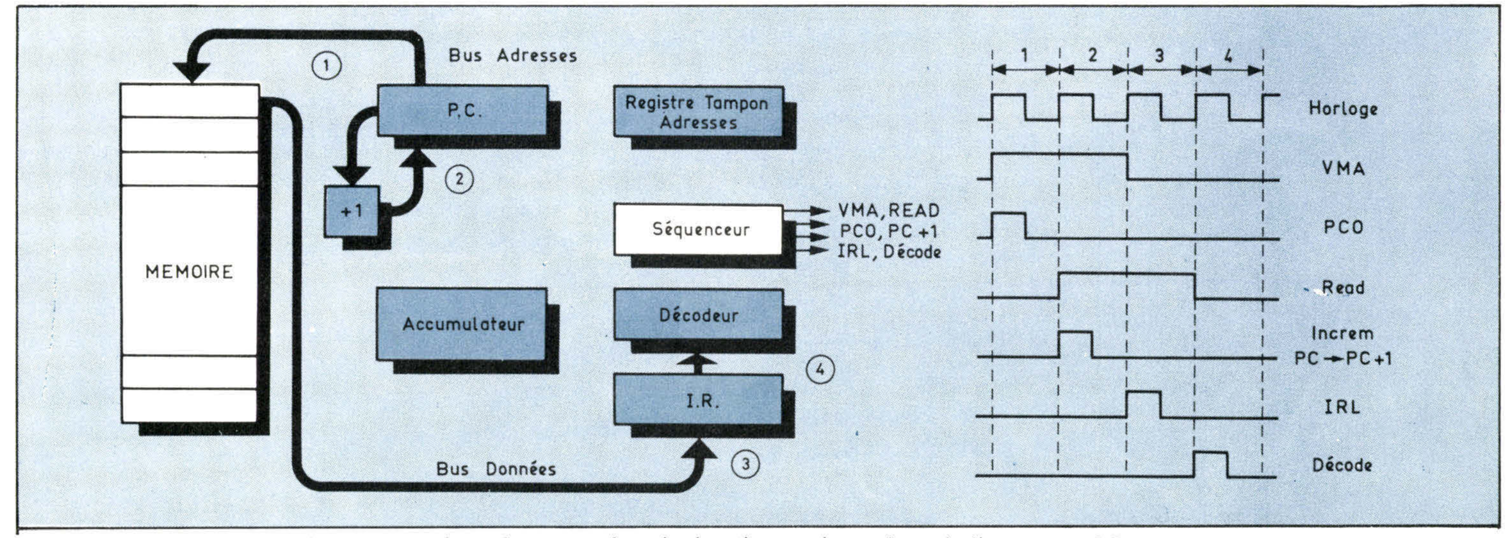


Fig. 6. – Chronogramme et cheminement des informations lors du deuxième cycle machine de l'instruction LDA. Ce cycle correspond à la recherche de la parte basse de l'adresse.

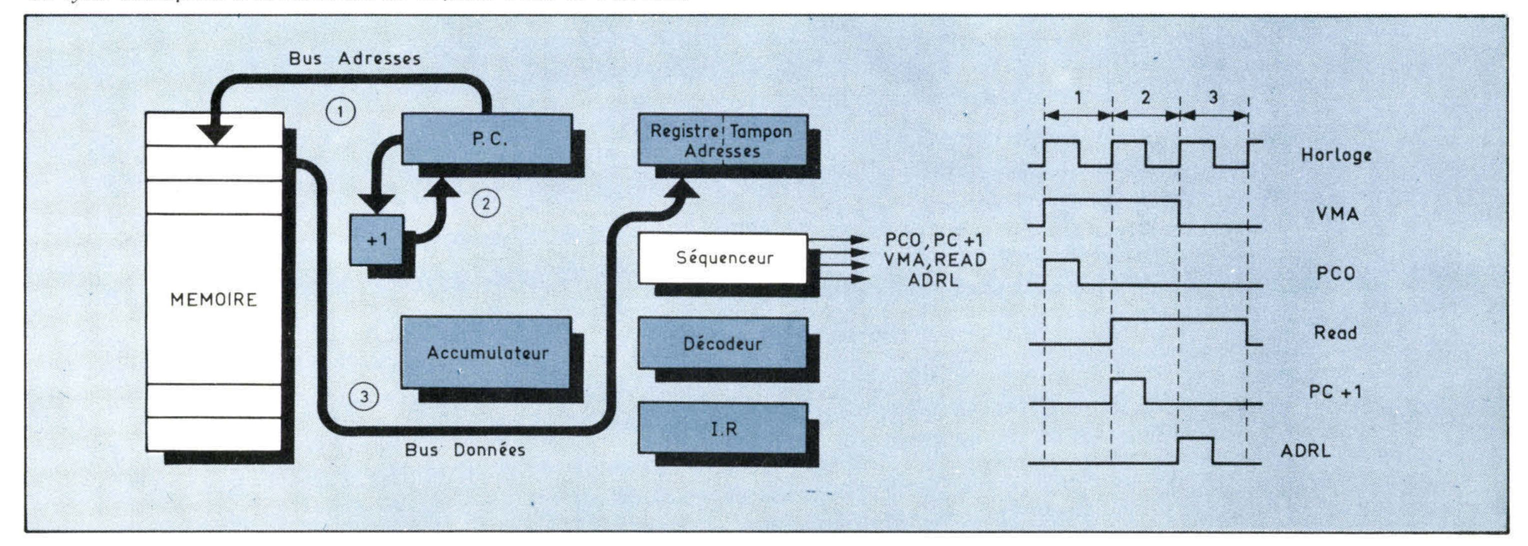
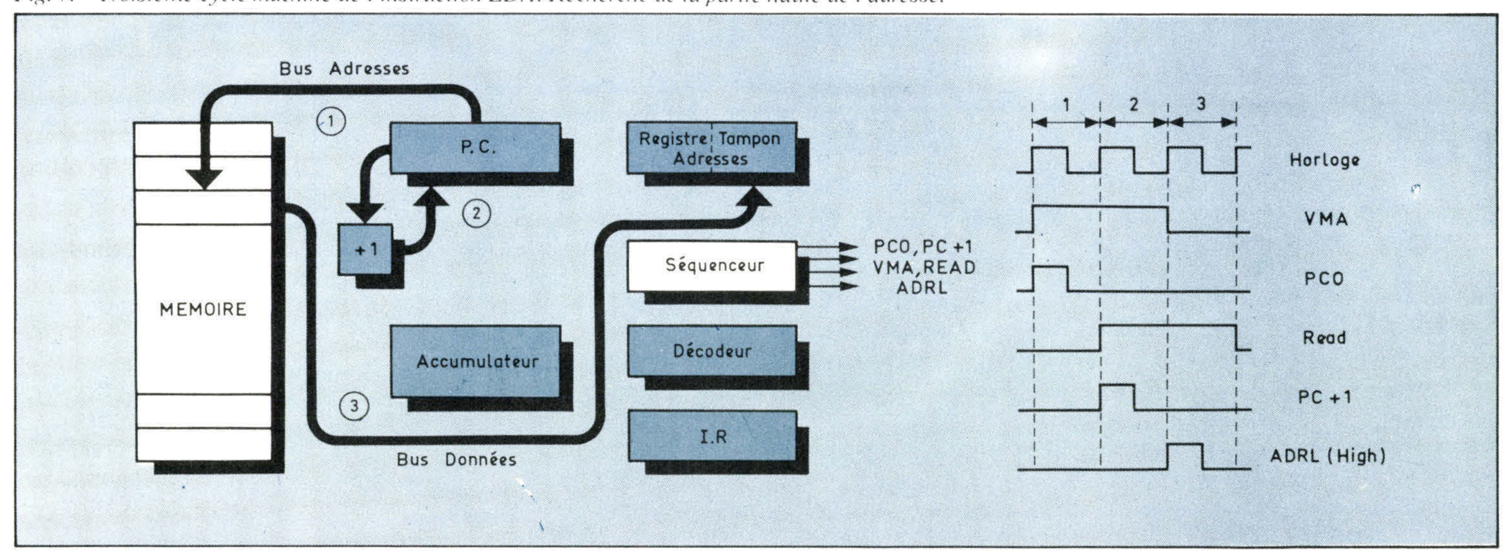


Fig. 7. – Troisième cycle machine de l'instruction LDA. Recherche de la partie haute de l'adresse.



Diriger de façon séquentielle les informations nécessite la connaissance successive des différentes adresses-mémoire.

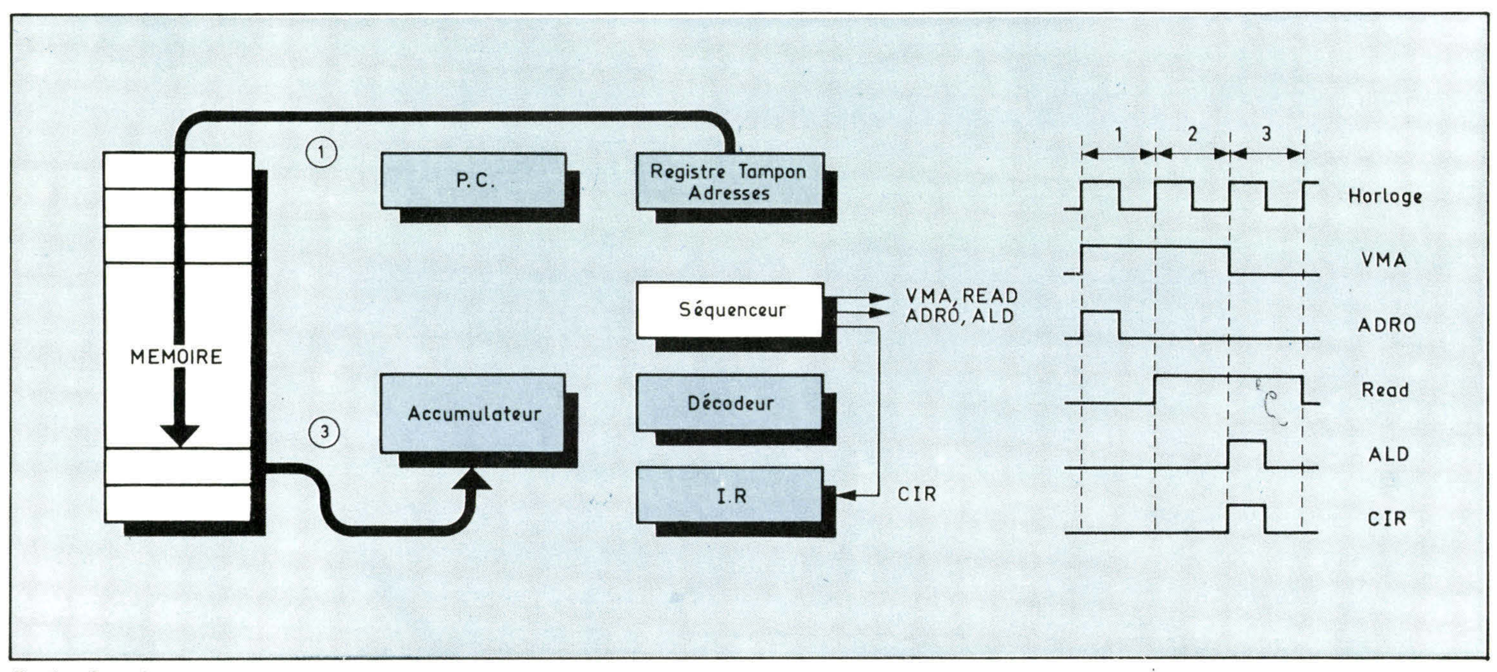


Fig. 8. – Quatrième cycle machine de l'instruction LDA. Chargement de la donnée dans l'accumulateur.

Fig. 9. – Micro instructions émises par le séquenceur pour réaliser l'instruction LDA (adresse).

Etude de STA (adresse)

1er cycle machine

A la première impulsion d'horloge le contenu du compteur ordinal est déposé sur la ligne adresse.

Pendant la deuxième impulsion d'horloge, la mémoire recherche son information et le compteur ordinal est incrémenté.

A la troisième impulsion d'horloge, le contenu du mot mémoire est transféré dans le registre d'instruction.

Pendant la quatrième impulsion d'horloge, l'information stockée dans le registre d'instruction est décodée.

2e cycle et 3e cycle machine

La configuration binaire de l'instruction reçue indique qu'il faut aller chercher en mémoire les deux octets suivants. Par conséquent, le deuxième et troisième cycles sont rigoureusement identiques à l'exemple précédent.

4e cycle

A la première impulsion d'horloge de ce cycle, le contenu du registre adresse temporaire est envoyé sur le bus adresse à l'aide d'une impulsion ADRO (ADdress Register Out) et nous expédions également l'information VMA. Pendant ce temps, l'information contenue dans l'accumulateur est envoyée dans le registre tampon à l'aide d'une instruction ACCO (ACCumulator Out).

5e cycle

A la première impulsion de ce cycle, l'information contenue dans le registre tampon est appliquée sur le bus des données. Il est envoyé en même temps un ordre d'écriture (write) à la mémoire.

A la deuxième impulsion d'horloge, l'information est écrite dans la mémoire et l'on envoie l'ordre de remettre à zéro, le registre d'instruction.

Les 4e et 5e cycles auraient pu être confondus en un seul cycle machine. Chez certains constructeurs, ils sont séparés pour indiquer que l'accumulateur est une mémoire RAM et il n'est pas possible de lire et d'écrire en même temps dans les RAM. Il est donc nécessaire dans un premier temps de transférer le contenu de l'accumulateur dans le registre tampon avant d'écrire cette information dans la mémoire principale. Dans ce cas, on considère le premier transfert comme un accès à la mémoire, donc un cycle machine.

Nous avons décrit ici le séquencement d'instructions d'un microprocesseur type, comme le 8080 de Intel par exemple. Tous les microprocesseurs n'ont pas rigoureusement le même mode de séquencement. Par exemple dans le M 6800* de Motorola qui utilise une horloge de fréquence plus faible (1 MHz au lieu de 2 MHz dans le 8080), le séquencement des informations se fait en utilisant les deux phases d'horloge et un décalage temporel (timing).

Si nous reprenons par exemple l'instruction LDA (adresse), nous aurons pour le premier cycle machine le diagramme de la **figure 10.**

Sur le front montant de Φ_1, on envoie l'ordre PCO ; c'est-à-dire que le contenu du compteur ordinal est déposé sur les lignes d'adresse. Dans un délai maximal de 300 nS apparaissent les informations VMA (Validation du bus d'adresse) et READ (ordre de lecture de la mémoire).

Sur le front montant de Φ_2, on envoie l'ordre PC + 1, c'est-à-dire que l'on incrémente le compteur programme. Pendant ce temps, la mémoire recherche son information qui devient disponible sur le bus de données.

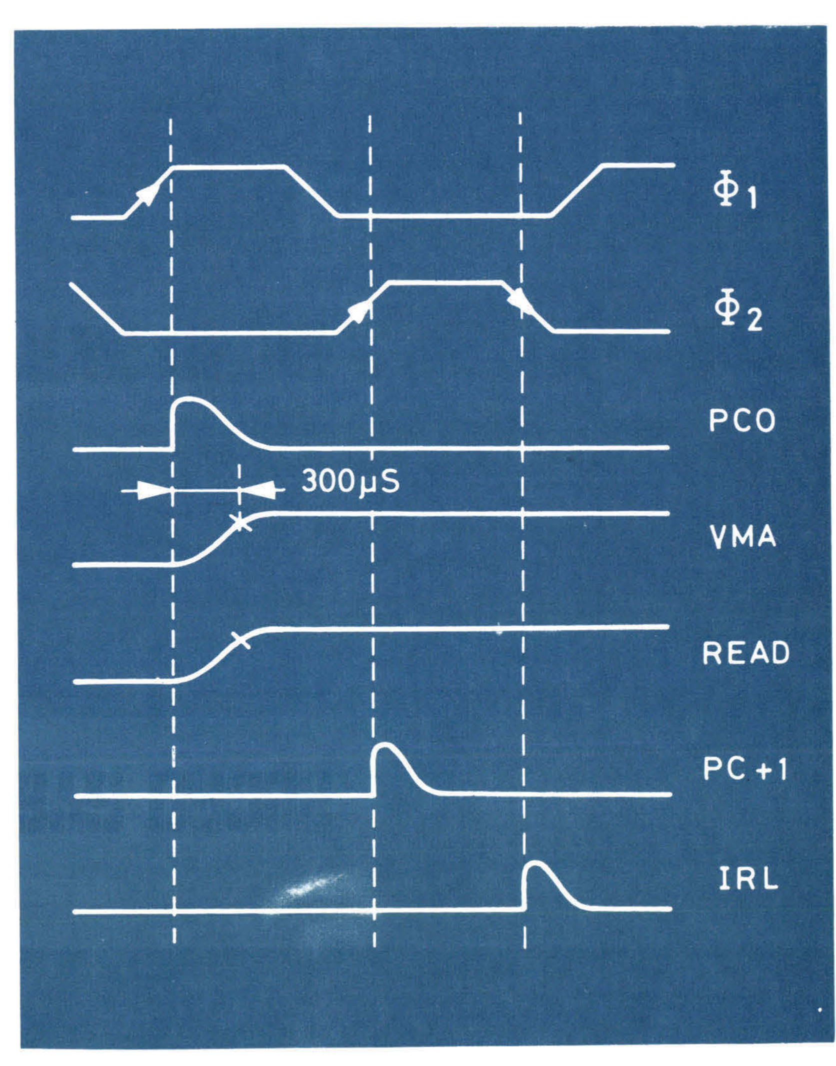


Fig. 10. – Séquencement du 1er cycle machine de l'instruction LDA pour le microprocesseur 6800. Noter l'action des 2 phases de l'horloge.

* *ou le SFF 96800 de Sescosem.*

Actuellement, les constructeurs doivent réaliser un compromis entre la vitesse d'exécution et la simplification de programmation des instructions.

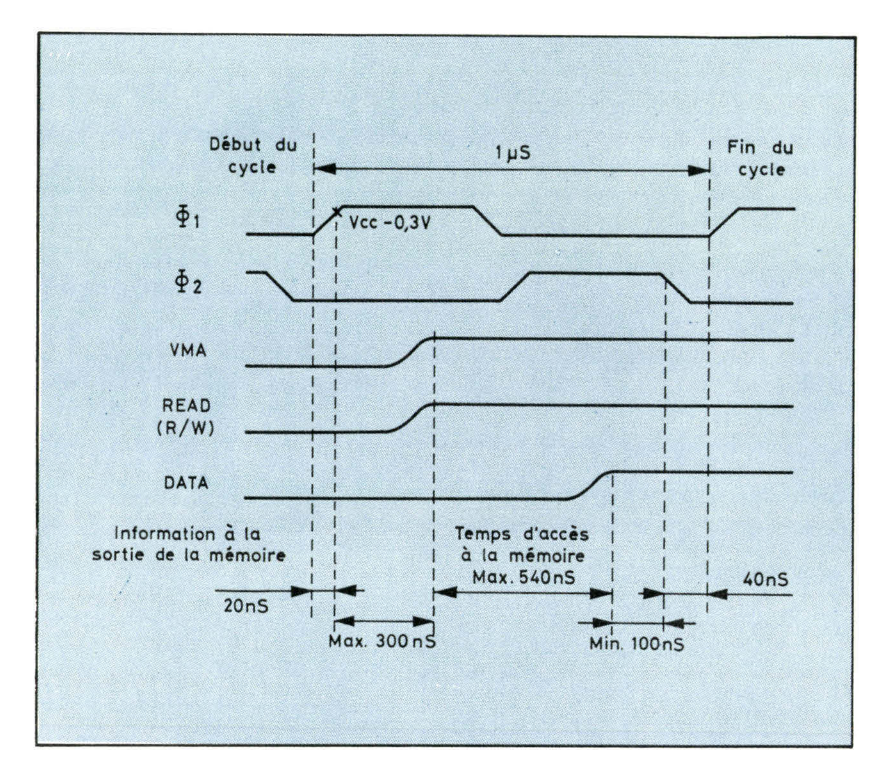


Fig. 11. – Lors d'un séquencement, il est nécessaire de respecter le « timing » indiqué par le constructeur.

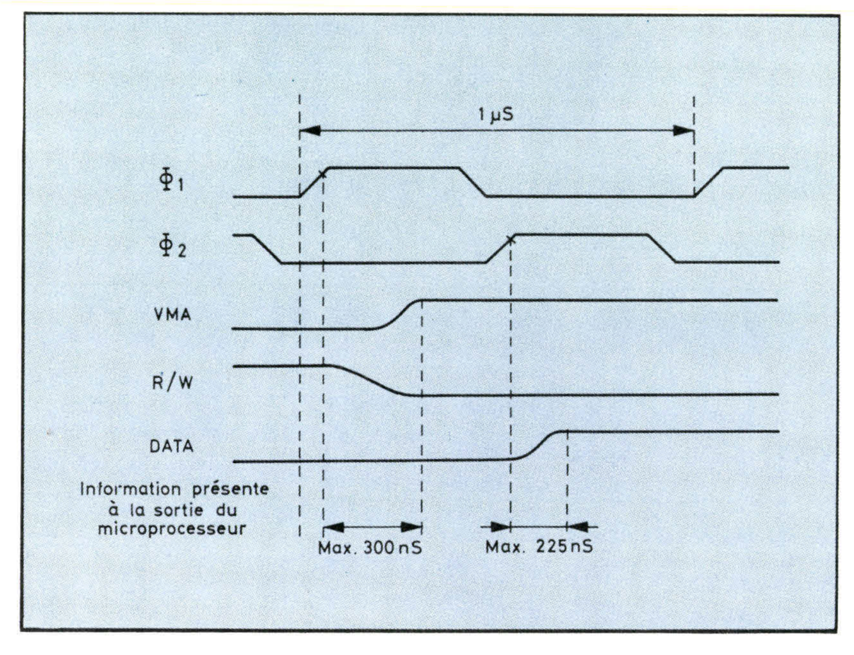


Fig. 12. – Chronogramme de l'inscription d'une information en mémoire.

Sur le front descendant de Φ_2 on envoie l'ordre IRL qui transfère l'information de la mémoire dans le registre instruction et l'on décode immédiatement l'information. Pour que ce cycle machine soit réalisé correctement, il est nécessaire de respecter le « timing » indiqué par le constructeur **(fig. 11).** L'inscription d'une information en mémoire est représenté sur le chronogramme de la **figure 12.**

Le principe d'accès à la mémoire pour les autres cycles machines exécutant l'instruction LDA (adresse) est identique au premier cycle décrit. Si ce système présente l'inconvénient d'avoir des signaux d'horloge bien calibrés et des mémoires ayant des temps d'accès inférieurs à ceux indiqués par le constructeur, il offre l'avantage d'un raisonnement simple pour l'utilisateur. En effet, on peut résumer le fonctionnement de la façon suivante : « Dans le M 6800 les adresses sont positionnées au début du front montant de Φ_1 et restent stables pendant tout le cycle ; les données sont disponibles en fin de Φ_2. Donc Φ_1 prépare les informations sur le bus de données, Φ_2 les récupère. » De plus dans ce cas, chaque cycle-machine a une durée égale à la période d'horloge, ce qui permettra de calculer facilement la durée totale d'un programme.

Sur ce simple exemple, il ne faut pas en conclure que ce système est plus rapide que celui étudié précédemment. Compte tenu des contraintes actuelles des microprocesseurs (nombre limité de broches, surface de la « puce » semi conductrice, technologie employée...), le constructeur est obligé de faire un compromis entre la vitesse d'exécution des instructions et la simplification de programmation offerte par ces instructions.

Le tableau ci-dessous indique pour le 8080 de Intel et le 6800 de Motorola, les valeurs extrêmes des durées des instructions et la valeur moyenne en faisant à partir de plusieurs exemples le rapport : durée totale du programme/nombre d'instructions.

CARACTERISTIQUES	8080	6800
Fréquence horloge	2 MHz	1 MHz
Durée minimale	2 µs	2 µs
Valeur moyenne	3,2 µs	4 µs
Durée maximale	9 µs	12 µs

Dans cet article, nous expliquerons plus particulièrement le cheminement des informations et le mode d'élaboration des micro-instructions du 8080, ce mode de fonctionnement étant le plus couramment rencontré à l'heure actuelle. Il est évident que les principes que nous étudierons seront transposables également à toutes les instructions des autres types de microprocesseurs.

Etude de JMP (adresse)

1er cycle machine

A la première impulsion d'horloge, le contenu du compteur ordinal est déposé sur la ligne adresse.

A la deuxième impulsion d'horloge, la mémoire recherche son information et le compteur ordinal est incrémenté.

A la troisième impulsion d'horloge, l'information mémoire est transférée dans le registre instruction.

A la quatrième impulsion d'horloge, l'information est décodée.

2e cycle machine

Identique aux exemples précédents.

3e cycle machine

Pour les trois premières impulsions d'horloge, le cycle est identique au précédent. A la quatrième impulsion d'horloge, le contenu du registre d'adresse temporaire est transféré dans le compteur ordinal.

En ce qui concerne ces instructions, deux faits importants sont à remarquer.

Lorsqu'il est nécessaire de générer un grand nombre de micro-instructions pour exécuter une instruction, la séquence de ces micro-instructions est presque toujours la même, du moins dans les premiers cycles machines, ce qui facilitera la réalisation de l'unité de commande.

En général, l'instruction nécessite plusieurs accès à la mémoire, ce qui augmente le nombre de cycles machines et par conséquent, ralentit le fonctionnement du microprocesseur. Afin d'augmenter la vitesse d'exécution des instructions, les constructeurs et les programmeurs, ont recherché des solutions d'adressage différentes, ayant pour but de réduire le nombre d'accès à la mémoire.

Examinons ensemble les principaux modes d'adressage couramment employés.

Les différents modes d'adressage

1 - Adressage implicite (inhérent)

L'instruction a une longueur de 1 octet. Elle est réalisée sur des valeurs contenues dans les registres internes de la machine. Le code de l'instruction et l'adresse des registres n'occupent qu'un seul mot de 8 bits.

Par exemple :

A B A = additionner le contenu des registres A et B
C L A = mettre à zéro le registre A
T A B = transférer la valeur de l'accumulateur A dans l'accumulateur B.

D'une façon générale, l'adressage implicite ne peut s'appliquer qu'aux registres internes du microprocesseur.

2 - Adressage immédiat

Après l'instruction suit immédiatement une valeur numérique sur un ou deux octets. Ceci est utilisé chaque fois que l'on aura une opération à effectuer sur une valeur constante. L'adressage immédiat est généralement représenté de la façon suivante :

L D A # $ 72

Cette expression signifie : charger l'accumulateur A (L D A) avec la valeur (#) hexadécimale ($) 72.

3 - Adressage direct

C'est un cas particulier de l'adressage normal de la mémoire où le positionnement des informations est restreint aux adresses 0 à 255, c'est-à-dire, à un seul octet.

4 - Adressage étendu (normal, absolu, extended)

L'instruction a une longueur de trois octets. Les deux derniers contiennent l'adresse absolue de l'opérande. C'est ce mode d'adressage que nous avons utilisé, dans les exemples d'instructions.

5 - Adressage indexé

Dans les programmes il est parfois nécessaire de se référer au **contenu d'un tableau en mémoire,** c'est-à-dire de se référer au nième mot d'une zone mémoire. Il est alors particulièrement avantageux de disposer d'un registre d'adressage spécial (registre d'index) qui contienne l'adresse de la base de ce tableau. Une instruction d'adressage indexé permet alors d'ajouter automatiquement un déplacement à ce registre d'index de manière à accéder au nième mot de la table, sans avoir à spécifier la base. Pratiquement, la plupart des microprocesseurs permettent d'accéder à une zone mémoire de 256 octets à partir de l'adresse définie par le registre d'index.

Soit, par exemple, à exécuter l'instruction :

L D A A 5, X

Cette instruction veut dire : charger l'accumulateur A avec le contenu de la cinquième position mémoire de la table dont le début est défini par le registre d'index IX.

La **figure 13** représente un exemple d'adressage indexé. L'adresse à rechercher est déterminée par l'adresse du registre d'index ajouté à l'adresse écrite dans l'instruction ; c'est-à-dire, dans l'exemple ci-dessus :

300 + 5 = 305.

6 - Adressage relatif

Cet adressage a pour but de faciliter les références à des adresses voisines de l'instruction en cours,

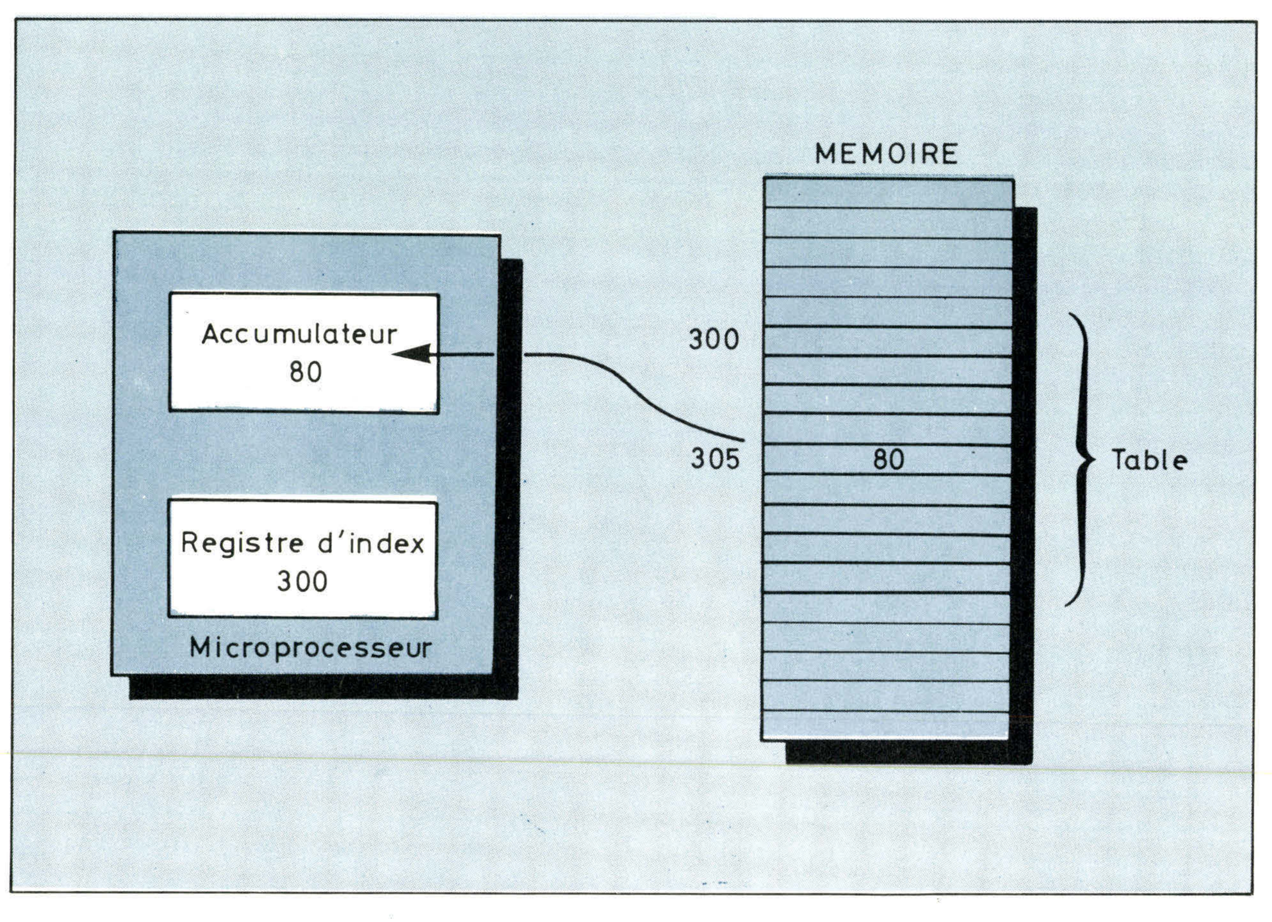


Fig. 13. – Exemple d'adressage indexé. Le registre d'index contient 300 si l'instruction à exécuter est LDA A5,X l'adresse utile sera 300 + 5 = 305.

en ajoutant au contenu du compteur ordinal, le déplacement relatif de la nouvelle adresse. La valeur étant contenue sur un octet, il est théoriquement possible d'accéder aux adresses CP – 128 à CP + 128. Ce mode d'adressage est pratiquement réservé aux instructions de branchement et de modification de données. Par exemple, si à l'adresse 1000 on trouve l'instruction BCC, qui veut dire : se brancher à une autre adresse si le bit de CARRY * est nul.

En adressage relatif lorsque l'on écrit BCC 80, dès que le bit de Carry du registre d'état est nul, le compteur programme passe automatiquement de l'adresse 1000 à l'adresse 1000 + 80 = 1080.

7 - Adressage indirect

Dans ce mode d'adressage, les octets suivants l'instruction ne contiennent pas l'adresse de l'information demandée, mais l'adresse d'un mot mémoire, où l'on trouvera l'adresse effective de l'information. Ce mode d'adressage peut être utilisé lorsqu'il faut partager une donnée entre plusieurs programmes. Sur certains microprocesseurs, il est utilisé après une interruption du fonctionnement ou lors de la mise en route.

Mnémonique	Types d'adressage	Code hexadécimal
LDA A	adressage immédiat	86
LDA A	adressage direct	96
LDA A	adressage indexé	A 6
LDA A	adressage normal	B 6

Remarques sur les différents types d'adressage

Le choix du mode d'adressage est fait par le programmeur, lors de la construction de son algorithme, en recherchant la vitesse d'exécution la plus élevée. Dans le code de l'instruction, les configurations binaires sont différentes suivant le mode d'adressage. Par conséquent, à la lecture de l'information le séquenceur après décodage, « sait » qu'il doit faire un accès mémoire, ou deux ou trois.

Exemple, l'instruction L D A A a pour le microprocesseur 6800 de Motorola, les codes hexadécimaux suivants :

Les microprocesseurs permettant l'adressage indexé, possèdent un registre supplémentaire, appelé registre d'index (Index Register I.X.). C'est un registre de 16 bits destiné à stocker une adresse. Cependant, il pourra être utilisé en programmation comme registre interne supplémentaire.

Jean FREMAUX *

** Carry : bit de retenue.*

** M. Jean Frémaux appartient à la direction générale de l'Ecole centrale d'électronique et assure les cours « microprocesseurs ».*

KINGDOM

Pour vous permettre d'étoffer votre bibliothèque de programmes, nous vous proposons ce mois-ci une application d'un type différent. Il s'agit toujours d'un jeu, mais cette fois, ce terme a un sens beaucoup plus large. Une appellation plus correcte serait : jeu d'entreprise. Car c'est une méthode d'enseignement largement répandue dans différentes sociétés.

Un sujet se trouve placé dans une situation particulière. Le problème est posé d'une façon nette. Il doit prendre lui-même plusieurs décisions, qui peuvent être plus ou moins bonnes. A chaque étape du jeu, il peut examiner les conséquences de ses décisions antérieures, confirmer les options choisies, ou rectifier sa ligne de conduite en modifiant certains paramètres qui ne paraissent pas satisfaisants.

Il n'y a donc pas de stratégie optimale. La situation évolue au cours du temps. Elle peut se dégrader ou s'améliorer en fonction d'événements accidentels extérieurs, à la volonté du sujet, mais surtout en fonction des choix effectués à chacune des étapes du jeu.



Le programme

Examinons plus en détails les particularités de ce programme tiré du livre « What to do after you hit return » *. Cet ouvrage contient une foule d'idées intéressantes sur de nombreux sujets concernant les micro-amateurs. Nous sommes placés dans les circonstances suivantes :

Un petit royaume, situé sur une île de 2 000 hectares, doit assurer sa survie. L'aspect économique est l'objet principal de ce jeu, car tout se ramène à des rallods (la monnaie en cours dans ce pays). Les revenus de l'Etat proviennent de différentes sources, en premier lieu, l'agriculture, car tous les habitants sont censés cultiver la terre. Cette terre peut aussi être vendue à des industriels, qui implantent leurs propres usines, et font venir la main-d'œuvre nécessaire à leur fonctionnement.

Enfin le tourisme peut être une source de revenus importante, à condition toutefois qu'il y ait un minimum d'industries, donc de commerce pour attirer des vacanciers.

Les dépenses que l'état doit effectuer chaque année sont de quatre types :

- La survie des habitants, en leur assurant un minimum de ressources de 100 rallods par an.
- Leur éducation, ce qui est profitable par la suite, car le rendement des cultures est directement fonction de ces dépenses.
- L'investissement nécessaire aux plantations.
- La lutte contre la pollution, car dès que des industries sont implantées, il y a des nuisances qui en découlent.

Tous ces facteurs sont donc à prendre en compte au début de chaque année, car nous jouons le rôle du roi qui décide seul et qui doit faire survivre son pays au moins huit ans. La répartition des crédits doit être effectuée correctement, car il faut pouvoir assurer ses ressources à court terme et à plus long terme.

La simulation a été poussée ici à un point remarquable, car nous allons voir de quelle manière ces facteurs influent les uns sur les autres :

- La surface cultivable est limitée. Chaque habitant ne peut travailler que 2 hectares.
- Le rendement des récoltes est lié aux sommes dépensées pour l'éducation des habitants pendant les trois dernières années, avec un plafond maximum de 10 rallods par personne.
- Ce rendement sera réduit au fur et à mesure que le niveau de pollution s'élève dans l'île.
- Cette pollution, comptée en unités, nécessite une somme de 0,44 rallods par unité pour être combattue.

* *Edité par Creative Computing*

● Elle est amenée par l'implantation d'industries qui, elles-mêmes, attirent le tourisme dans une certaine mesure. Il faut au minimum 2 hectares d'industrie pour voir débarquer les premiers estivants, et ces arrivées augmenteront jusqu'à ce que les usines occupent 26 hectares. Mais gare à la pollution qui les fera fuir.

● Les industriels suppléent eux-mêmes aux besoins de leurs propres travailleurs immigrés. Tout se passe bien dans l'île tant que le pourcentage de travailleurs immigrés ne dépasse pas 50 % de la population totale.

● La vente de terrain aux industriels rapporte entre 95 et 105 rallods par hectare, puis 44 rallods de taxes pendant chacune des années suivantes.

● L'île, qui dispose au début de 500 habitants environ, peut voir arriver de nouveaux colons, à condition que la vie paraisse agréable, c'est-à-dire que le revenu par habitant soit suffisamment élevé. Au contraire, si l'austérité est trop grande, plusieurs personnes quitteront l'île.

D'autres incidents peuvent se produire (en général désagréables), si les décisions du roi amènent des résultats impopulaires, ou si des fautes graves de gestion sont détectées.

Essayons de faire une partie, et voyons comment évolue la situation pendant quelques années **(tableau 1).**

BONNE CHANCE POUR VOS HUIT ANNEES

VOUS AVEZ 69 645 RALLODS, 501 HABITANTS ET 2 000 HECTARES
CETTE ANNEE, LES INDUSTRIELS VOUS OFFRENT 95 RALLODS PAR HECTARE ET LES PLANTATIONS VOUS COUTENT 12 RALLODS/HECTARE.

COMBIEN D'HECTARES VOULEZ-VOUS VENDRE AUX INDUSTRIELS ? **10**
COMBIEN DE RALLODS VOULEZ-VOUS DISTRIBUER A VOS HABITANTS ? **52 000**
COMBIEN D'HECTARES VOULEZ-VOUS PLANTER ? **990**
COMBIEN VOULEZ-VOUS DEPENSER POUR L'EDUCATION ? **5 010**
QUEL BUDGET ACCORDEZ-VOUS A LA LUTTE CONTRE LA POLLUTION ? **100.**

12 TRAVAILLEURS IMMIGRES SONT ARRIVES
4 HABITANTS SONT VENUS S'INSTALLER
VOUS AVEZ RECOLTE 990 HECTARES
CEUX-CI VOUS RAPPORTENT 39 600 RALLODS
LE TOURISME VOUS A RAPPORTE 2 956 RALLODS
LES TAXES PRELEVEES AUX INDUSTRIELS VOUS RAPPORTENT 440 RALLODS.

VOUS AVEZ 56 041 RALLODS, 505 HABITANTS, 12 TRAVAILLEURS IMMIGRES ET 1 990 HECTARES
CETTE ANNEE, LES INDUSTRIELS VOUS OFFRENT 100 RALLODS PAR HECTARE ET LES PLANTATIONS VOUS COUTENT 10 RALLODS/HECTARE.

COMBIEN D'HECTARES VOULEZ-VOUS VENDRE AUX INDUSTRIELS ? **10**
COMBIEN DE RALLODS VOULEZ-VOUS DISTRIBUER A VOS HABITANTS ? **50 500**
COMBIEN D'HECTARES VOULEZ-VOUS PLANTER ? **980**
COMBIEN VOULEZ-VOUS DEPENSER POUR L'EDUCATION ? **5 050**
QUEL BUDGET ACCORDEZ-VOUS A LA LUTTE CONTRE LA POLLUTION ? **300.**

20 TRAVAILLEURS IMMIGRES SONT ARRIVES
1 HABITANT A QUITTE L'ILE
VOUS AVEZ RECOLTE 980 HECTARES
CEUX-CI VOUS RAPPORTENT 43 120 RALLODS
LE TOURISME VOUS A RAPPORTE 6 410 RALLODS
LES TAXES PRELEVEES AUX INDUSTRIELS VOUS RAPPORTENT 880 RALLODS.

VOUS AVEZ 50 721 RALLODS, 504 HABITANTS, 32 TRAVAILLEURS IMMIGRES ET 1 980 HECTARES
CETTE ANNEE LES INDUSTRIELS VOUS OFFRENT 97 RALLODS PAR HECTARE ET LES PLANTATIONS VOUS COUTENT 14 RALLODS/HECTARE.

COMBIEN D'HECTARES VOULEZ-VOUS VENDRE AUX INDUSTRIELS ? **10**
COMBIEN DE RALLODS VOULEZ-VOUS DISTRIBUER A VOS HABITANTS ? **50 000**
COMBIEN D'HECTARES VOULEZ-VOUS PLANTER ? **970**
COMBIEN VOULEZ-VOUS DEPENSER POUR L'EDUCATION ? **5 040**
MAIS VOUS N'AVEZ PLUS QUE 1 691 RALLODS !
COMBIEN VOULEZ-VOUS DEPENSER POUR L'EDUCATION ? **1 191**
QUEL BUDGET ACCORDEZ-VOUS A LA LUTTE CONTRE LA POLLUTION ? **500**

4 PERSONNES SONT MORTES DE MISERE
VOUS ETES OBLIGE DE DEPENSER 36 RALLODS POUR LES ENTERRER
22 TRAVAILLEURS IMMIGRES SONT ARRIVES
2 HABITANTS ONT QUITTE L'ILE
ETC.

Tableau I. – Vous êtes le monarque d'un petit royaume et vous devez faire survivre votre pays au moins 8 ans. Après ces huit années (en réalité quelques dizaines de minutes), la machine énoncera les conséquences de votre gestion sur l'économie du pays.

Il est facile de voir les avantages d'une telle méthode de simulation. Des décisions dont les effets ne commencent à apparaître qu'à long terme, se déroulent ici en continu. Plusieurs tentatives peuvent être faites sur machine, avant d'adopter l'une ou l'autre voie dans la réalité. L'initiation à la gestion est présentée sous une forme plus vivante que les cours magistraux !

Ici le divertissement rejoint des besoins plus sérieux.

Bien sûr, le domaine d'application de ce type de programme n'est pas limité. La gestion d'une entreprise peut se simuler d'une manière tout à fait identique. Le nombre de paramètres qui intervient peut être étendu au gré de l'imagination de chacun.

La taille mémoire disponible devient alors une contrainte. Ce programme nécessite déjà 7 k-octets en plus de la zone réservée au Basic. Ce volume important est dû en partie aux nombreux commentaires qui agrémentent le jeu.

Voici donc en Basic, le listing qui vous permettra d'exercer vos talents de monarque, mais chargé de gouverner un grand nombre de ressortissants violents.

**A. Seligman
et H. Eymard-Duvernay**

LISTING DU PROGRAMME

```
10   REM KINGDOM
20   REM CETTE VERSION PAR ANDRE SELIGMAN ET HUGUES EYMARD
30   N5 = 8
40   N4 = 8
50   PRINT
60   PRINT « BONNE CHANCE POUR VOS » ; N4 ; « ANNEES »
400  M = 69 000 + RND * 2 000
405  M = INT (M)
410  P = INT (490 + RND * 20)
420  L = 2 000
430  W = 0 : X2 = 0 : E0 = 0 : X1 = 0
440  E1 = 0 : S9 = 0 : S0 = 0 : M5 = 0 : S1 = 0
445  D1 = INT (P/3)
450  L1 = INT (RND * 5 + 10)
460  L0 = INT (RND * 10 + 95)
470  GOSUB 2 220
480  PRINT « VOUS AVEZ » ; M ; « RALLODS, » ;
490  PRINT P ; « HABITANTS, »
500  IF W = 0 THEN 520
510  PRINT W ; « TRAVAILLEURS IMMIGRES » ;
520  PRINT « ET » ; L ; « HECTARES »
530  PRINT « CETTE ANNEE, LES INDUSTRIELS VOUS OFFRENT » ;
535  PRINT L0 ; « RALLODS PAR HECTARE »
540  PRINT « ET LES PLANTATIONS COUTENT » ;
550  PRINT L1 ; « RALLODS/HA. »
555  IF S1 = 0 THEN 570
560  PRINT « LE NIVEAU DE POLLUTION EST DE » ; S1 ; « UNITES ».
570  GOSUB 2 220
580  PRINT « COMBIEN D'HA VOULEZ-VOUS VENDRE AUX INDUSTRIELS » ;
590  INPUT L2
600  IF L2 < 0 THEN 580
610  IF L2 < (L - 1 000) THEN 680
620  PRINT « ALLONS ! VOUS N'AVEZ QUE » ; L - 1 000 ;
625  PRINT « HECTARES DE TERRE DISPONIBLES »
630  IF X1 <> 0 THEN 580
640  PRINT « (LES INDUSTRIELS NE SONT PAS DISPOSES A ACHETER DE LA FORET »
650  PRINT « DU FAIT DES COUTS D'ABATTAGE) »
660  X1 = 1
670  GOTO 580
680  M = INT (M + L2 * L0)
690  L = INT (L - L2)
700  X9 = INT (44 * (2 000 - L))
710  M2 = 0
```

```
720   M3 = 0
730   M4 = 0
740   PRINT « COMBIEN DE RALLODS VOULEZ-VOUS DISTRIBUER » ;
750   PRINT « A VOS HABITANTS » ;
760   INPUT M1
770   IF M1 < 0 THEN 740
780   IF M1 < = M THEN 810
790   GOSUB 2250
800   GOTO 740
810   M = INT (M - M1)
820   IF M = 0 THEN 1 170
830   PRINT « COMBIEN D'HECTARES VOULEZ-VOUS PLANTER » ;
840   INPUT L3
850   IF L3 < 0 THEN 830
860   IF L3 < = 2 * P THEN 890
870   PRINT « CHAQUE HABITANT NE PEUT PLANTER QUE 2 HA »
880   GOTO 950
890   IF L3 < = L - 1 000 THEN 920
900   PRINT « VOUS N'AVEZ QUE » ; L - 1 000 ; HA DE TERRE CULTIVABLE
910   GOTO 950
920   M4 = INT (L3 * L1)
930   IF M4 < = M THEN 990
940   GOSUB 2 250
950   PRINT « VOTRE BUDGET VOUS LIMITE A » ; INT (M/L1) ; « HA »
960   PRINT « LA POPULATION PEUT TRAVAILLER » ; P * 2 ; « HA »
970   PRINT « ET LA SURFACE DISPONIBLE EST DE » ; L - 1 000 ; « HA »
980   GOTO 830
990   M = INT (M - M4)
1000  IF M = 0 THEN 1170
1010  PRINT « COMBIEN VOULEZ-VOUS DEPENSER POUR L'EDUCATION » ;
1020  INPUT M2
1030  IF M2 < 0 THEN 1010
1040  IF M2 < = M THEN 1070
1050  GOSUB 2250
1060  GOTO 1010
1070  M = INT (M - M2)
1080  IF M = 0 THEN 1170
1090  IF L = 2000 THEN 1170
1100  PRINT « QUEL BUDGET ACCORDEZ-VOUS A LA LUTTE CONTRE LA POLLUTION » ;
1110  INPUT M3
1120  IF M3 < 0 THEN 1100
1130  IF M3 < = M THEN 1160
1140  GOSUB 2250
1150  GOTO 1100
1160  M = INT (M - M3)
1170  GOSUB 2220
1180  D0 = 0
1190  E2 = M2/P
1200  D2 = INT (P - M1/100)
1210  IF D2 < = 0 THEN 1240
1220  D0 = D2
1230  PRINT D2 ; « HABITANTS SONT MORTS DE MISERE »
1240  D3 = INT ((S1/100 000) * (. 1 + . 4 * RND) * P)
1250  IF D3 < = 0 THEN 1280
1260  PRINT D3 ; « HABITANTS SONT MORTS PAR EXCES DE POLLUTION »
1270  D0 = D0 + D3
1280  IF DO < = 0 THEN 1400
1290  F = D0 * 9
1300  PRINT « VOUS ETES OBLIGE DE DEPENSER » ; F ; « RALLODS »
1305  PRINT « POUR LES ENTERRER »
1310  M = INT ( M - F)
1320  IF M < = 0 THEN 1400
1330  PRINT « VOTRE FAIBLE BUDGET VOUS OBLIGE A VENDRE DU TERRAIN »
1340  T0 = - INT (M/L0) + 1
1350  L = L - T0
```

```
1360  IF L > = 1000 THEN 1390
1370  PRINT « VOUS N'AVEZ PAS ASSEZ DE TERRAIN »
1380  GOTO 2270
1390  M = M + T0 * L0
1400  IF D0 > 200 THEN 2300
1410  P = P - D0
1420  D1 = D1 - D0
1430  IF D1 < 0 THEN 2410
1440  IF D2 < = 2 THEN  1460
1450  IF M > 500 THEN 2520
1460  IF L2 = 0 THEN 1520
1470  T0 = INT (L2 + 2 * L2 * RND)
1480  IF W = 0 THEN 1500
1490  T0 = INT (T0 + . 1 * W)
1500  PRINT TO ; « TRAVAILLEURS IMMIGRES SONT ARRIVES »
1510  W = W + T0
1520  T0 = INT (((500 - P)/10 - D3/3 - D2/5) * . 75 * (1 + RND))
1530  PRINT ABS (T0) ; « HABITANTS » ;
1540  IF T0 < 0 THEN 1570
1550  PRINT « SONT VENUS S'INSTALLER . »
1560  GOTO 1590
1570  PRINT « ONT QUITTE L'ILE . »
1590  P = P + T0
1600  IF P < W THEN 2480
1610  T0 = 0
1620  IF L = 2000 THEN 1680
1630  T0 = INT ((S1/100 000) * L3)
1640  IF T0 < = L3 THEN 1660
1650  T0 = L3
1660  IF T0 = 0 THEN 1680
1670  PRINT « SUR VOS » ; L3 ; « HECTARES PLANTES , »
1680  PRINT « VOUS AVEZ RECOLTE » ; L3 - T0 ; « HA »
1690  IF T0 = 0 THEN 1760
1700  IF X2 > = 2 THEN 1760
1710  PRINT « (A CAUSE DE » ;
1720  IF X2 = 0 THEN 1740
1730  PRINT « L'AUGMENTATION DE » ;
1740  PRINT « LA POLLUTION ) »
1750  X2 = X2 + 1
1760  T1 = INT ((39 + RND * 20) * (1 + . 25 * (E0 + E1) / 20))
1770  PRINT « CEUX-CI VOUS RAPPORTENT » ; INT (T1 * (L3 - T0)) ; « RALLODS . »
1780  M = M + INT (T1 * (L3 - T0))
1790  E0 = E1
1800  IF E2 < = 10 THEN 1820
1810  E2 = 10
1820  E1 = E2
1830  T0 = 2000 - L
1840  IF T0 < 2 THEN 2100
1850  IF T0 < = 26 THEN 1870
1860  T0 = 26
1870  T0 = INT (T0 * 500 * (. 52 + . 5 * RND))
1880  T1 = S1/100 000
1890  IF T1 < = 1 THEN 1910
1900  T1 = 1
1910  T1 = T1 * T0
1920  PRINT « LE TOURISME VOUS A RAPPORTE » ; INT (T0 - T1) ; « RALLODS »
1930  M = M + INT (T0 - T1)
1940  IF INT (T0 - T1) < = M5 THEN 2080
1950  IF S1 < = S9 THEN 2080
1960  PRINT « EN NETTE DIMINUTION, CAR »
1970  T2 = INT (RND * 5 + 1)
1980  ON T2 GOTO 1990, 2010, 2030, 2050, 2070
1990  PRINT « DE NOMBREUX POISSONS SONT MORTS »
2000  GOTO 2080
2010  PRINT « LES PETITS OISEAUX CHANTENT DE MOINS EN MOINS »
```

```
2020 GOTO 2080
2030 PRINT « L'EAU SALIT LES BAINS DES CURISTES »
2040 GOTO 2080
2050 PRINT « LES MAUVAISES ODEURS DECOURAGENT LES PLAISANCIERS »
2060 GOTO 2080
2070 PRINT « LES HOTELS PARAISSENT NOIRS A CAUSE DES FUMEES »
2080 M5 = INT (T0 - T1)
2090 S9 = S1
2100 S2 = (2000 - L) * (2000 - L) - M3/.44
2110 PRINT « LES TAXES PRELEVEES AUX INDUSTRIELS » ;
2115 PRINT « VOUS RAPPORTENT » ; X9 ; « RALLODS »
2120 IF S2 > = 0 THEN 2170
2130 S1 = INT (S1 + S2/2)
2140 IF S1 > = S0 THEN 2160
2150 S1 = S0
2160 GOTO 2190
2170 S1 = INT (S1 + S2)
2180 S0 = INT (S0 + S2/10)
2190 N5 = N5 - 1
2200 IF N5 > 0 THEN 450
2210 GOTO 2570
2220 PRINT
2230 PRINT
2240 RETURN
2250 PRINT « MAIS VOUS N'AVEZ PLUS QUE » ; M ; « RALLODS ! »
2260 RETURN
2270 PRINT « VOUS AVEZ ETE RENVERSE, ET MALHEUREUSEMENT »
2280 PRINT « VOUS VOUS RETROUVEZ EN PRISON »
2290 STOP
2300 PRINT D0 ; « PERSONNES SONT MORTES CETTE ANNEE !!! »
2310 PRINT « A CAUSE DE VOS GRAVES NEGLIGENCES »
2315 PRINT « NON SEULEMENT VOUS AVEZ ETE RENVERSE »
2320 PRINT « MAIS DE PLUS » ;
2330 T0 = INT (RND * 3 + 1)
2340 ON T0 GOTO 2350, 2370, 2390
2350 PRINT « ON VOUS A CREVE LES YEUX . »
2360 STOP
2370 PRINT « VOUS ETES L'ENNEMI PUBLIC NUMERO UN . »
2380 STOP
2390 PRINT « VOUS AVEZ DROIT AU GOUDRON ET AUX PLUMES . »
2400 STOP
2410 PRINT « PLUS D'UN TIERS DE LA POPULATION EST MORT DEPUIS »
2420 PRINT « QUE VOUS AVEZ ETE ELU . LES MALHEUREUX SURVIVANTS »
2430 PRINT « VOUS ONT LITTERALEMENT ETRIPE . »
2450 STOP
2480 PRINT « LES TRAVAILLEURS IMMIGRES SONT EN MAJORITE . »
2490 PRINT « ILS SE SONT REVOLTES ET ONT PRIS LE POUVOIR . »
2500 GOTO 2270
2520 PRINT « VOUS AVEZ ENCORE ASSEZ D'ARGENT EN CAISSE . »
2530 PRINT « OR PLUSIEURS PERSONNES SONT MORTES DE MISERE »
2540 PRINT « LA POPULATION S'EST SOULEVEE ET VOUS A »
2550 PRINT « ACCULE AU SUICIDE . »
2560 STOP
2570 PRINT
2580 PRINT « FELICITATIONS !! »
2590 PRINT « VOUS AVEZ ACCOMPLI CORRECTEMENT VOTRE MANDAT »
2600 PRINT « DE » ; N4 ; « ANNEES. VOUS POUVEZ VOUS REPRESENTER »
2610 PRINT « AUX PROCHAINES ELECTIONS . »
2620 PRINT « TAPEZ 0 SI VOUS LE VOULEZ » ;
2630 INPUT Q
2640 N5 = N4
2650 IF Q = 0 THEN 440
2660 STOP
2670 END
```

Les clubs de micro-informatique au Sicob-1978

Cette année le Salon de l'Informatique et de l'Organisation de Bureau présentait une grande nouveauté par rapport aux années précédentes : face à une expansion continue il s'est étendu sur le Parvis de la Défense, à l'extérieur de la « Boule ». Signe des temps, on y trouvait des systèmes informatiques en « pur acajou », chers, et en « formica » bien solide et cinq fois moins cher, pouvant abattre la même besogne, dans le style « à programmer soi-même », en lisant les revues et les livres de programmation. Les derniers systèmes occupaient principalement la « Sicob-Boutique », sur le Parvis.

Qui dit programmation propre et systèmes informatiques bon marché, dit regroupement des gens ayant les mêmes intérêts, bourse d'échange d'idées et de programmes, formation, etc. et l'on arrive au concept de Club ou d'Association de micro-informatique. Trois clubs publics étaient présents : AFIN-Cau, OEDIP et AFMI ainsi qu'un quatrième, dont on apprenait l'existence par un papier sur le stand de la Mission pour l'Informatique : Microtel, dont le siège social se trouve au CNET à Issy les Moulineaux mais paraissant de ce fait un peu moins accessible au public

Tous les clubs présents affichent le même désir de promouvoir la Micro-Informatique tout en conservant certaines particularités :

AFIN-Cau, 54, rue Saint-Lazare, 75009 Paris, tél. : 280-17-88

Il s'agit de la section Constructeurs Amateurs Utilisateurs de l'Association Française des Informaticiens. Ils disposent du secrétariat de l'AFIN et se réunissent, en principe, tous les mardi soir, vers 18 heures, au 7, rue Poulletier dans l'Ile Saint-Louis (salle au sous-sol). Leurs réalisations, présentes sur le stand comportaient une mémoire à accès multiple, que nous avions déjà vue à Micro-Expo, au mois de mai, et une plaque kit d'évaluation avec laquelle Dominique Bultez crée de la musique, dans la tradition des boîtes à musique à signal carré.

Les panneaux graphiques parlaient de connexion de boîtiers, de schémas de systèmes à micro-processeurs. Le club est donc en mesure de former les adhérents à l'électronique des systèmes, contrairement à l'AFIN qui regroupe des programmeurs par excellence. L'inscription et participation aux réunions sont ouvertes à tout le monde, pour un montant de cotisation modeste : 170 F/an (200 F chez OEDIP et AFMI)

OEDIP, 8, place Sainte-Opportune, 75001 Paris. Tél. : 508-46-21.

Ce deuxième club nous a semblé orienté « Soft », vers la programmation des micro-systèmes. Sur leur stand, le plus apaisant de la « Boutique », on sentait une démarche vers l'homme de la rue. Les panneaux en écriture et dessins d'enfants contribuaient à cette impression. A côté d'un Apple-II doté d'une mini-disquette avec la série de programmes de présentation habituelle de la firme d'outre-océan, figurait un micro-ordinateur « production propre », le Norodata. Un deuxième stand, sous la coupole, en exposait un second exemplaire, ce qui laisse croire à une vocation commerciale de ce micro-ordinateur, muni d'un Basic, d'un écran T.V. et d'un clavier, dans la meilleure tradition des micro-ordinateurs actuels. Nous lui souhaitons bon vent ! Il constitue, certainement, le meilleur matériel didactique des membres du club.

AFMI, 75, boulevard Saint-Germain, 75005 Paris, Tél. : 844-77-81

Un panneau « Des Réalisateurs Audacieux » annonçait leur stand. Ce club nous paraît très complet : ses vocations couvrent une large gamme, allant de la formation permanente (ils ont un numéro de convention) jusqu'aux réalisations complètes où l'on présente un produit fini, à microprocesseur « caché ».

Nous avons trouvé sur leur stand une richissime documentation au sujet de leurs séminaires et des réalisations : un clavier syllabique, commandant une machine IBM I/O à boule, encodé par microprocesseur et un tapis, réalisé par un tisserand micro-amateur. Agréables, ces réalisations complètes, presque industrielles. Elles dénotent une bonne organisation, et donnent une réponse positive à la question de savoir si la micro-informatique est un gadget de luxe inutile, ou un facteur de vie plus commode et plus agréable. Le tapis du tisserand, réalisé en 4 bits, sur 1/2 port de PIA, présentait un modèle, conçu par micro-ordinateur, des plus jolis. Sur ce même stand on constatait que ce club est en liaison avec d'autres clubs, car un dessin par ordinateur portait en légende l'adresse de l'**ADAO, Association pour le Développement de l'Art par Ordinateur, 97, boulevard Maussard, 21 100 Dijon.** Paradoxalement, l'accès au club n'est pas « immédiat ». Les nouveaux membres devant faire preuve de formation sous forme d'une sorte de stage, aidés par des « parrains », avant une « titularisation » complète. L'Association offre comme matériel de cours de formation permanente un micro-ordinateur à monter soi-même (16 k de RAM et BASIC à 9 chiffres significatifs en flottant).

Microtel-club

Une documentation sommaire nous apprend que ce club d'amateurs de micro-informatique est lié à l'Administration des PTT et au CNET.

Leur tract donne un numéro de téléphone : monsieur Rinaudo 566-37-38. Par téléphone nous avons appris qu'il se trouve à Issy-les-Moulineaux, au siège du CNET.

Nous publierons de plus amples informations sur les Associations de micro-informatique dans nos prochains numéros.

André DORIS
le 30 septembre 1978

Un nouveau venu, dans le monde de la micro-informatique française au SICOB-1978

Proteus International a choisi une option, peut-être audacieuse à première vue, mais qui risque de s'avérer, en définitive, la plus sage. La direction du groupe, à la suite d'études peut-être les plus complètes sur le devenir du marché mondial de la micro-informatique, est arrivée à la conclusion que ce qui l'intéressait dans ce marché était l'utilisateur professionnel et le hobbyist avancé.

D'abord, c'est un système complet : jusqu'ici le micro-informaticien devait acheter son unité centrale chez X, son moniteur vidéo chez Y, et son imprimante – lorsqu'il en avait besoin – chez Z. Ici, rien de tout cela, sous la même marque, et chez les mêmes distributeurs, un système complet qui commence par le micro-ordinateur Proteus III et se complète au fur et à mesure de vos besoins, d'un moniteur, d'une imprimante et de mini-floppy.

Le matériel est proposé, au même prix, en beige clair, et en noir mat. On peut aussi, pourquoi pas, combiner un élément noir avec des éléments beige, puisqu'ils sont vendus séparément.

L'unité centrale micro-ordinatrice c'est Proteus III, qui a donné son nom au système. Il est proposé en deux versions, l'une 16-K et l'autre 32-K. Une innovation intéressante, et très sympathique, sur le plan commercial, réside dans le fait que le fabricant s'engage à vous faire passer en 32-K, si vous aviez commencé par du 16-K, en ne vous facturant que la différence du prix de tarif. Cela méritait d'être signalé.

Le clavier est un clavier standard de machine à écrire, ce qui ravira sûrement tous les utilisateurs et utilisatrices. Les touches de fonction sont au nombre de quatre, à gauche, bien au-dessus du clavier, et très clairement identifiées. Toutes les commandes sont ainsi face à l'utilisateur.

La visu s'appelle **Proteus Monitor.** Le constructeur a préféré, tout de suite, offrir ce qui se fait de mieux, un écran de 30 cm, de haute résolution, une lecture claire, lisible, reposante.

Proteus Print, l'imprimante est de type imprimante de bureau, travaillant sur 80 colonnes, à 1 200 Bauds (120 caractères/seconde). Un détail extrêmement intéressant : Proteus Print travaille sur papier normal, ce qui permet à l'acquéreur d'utiliser ses propres documents.

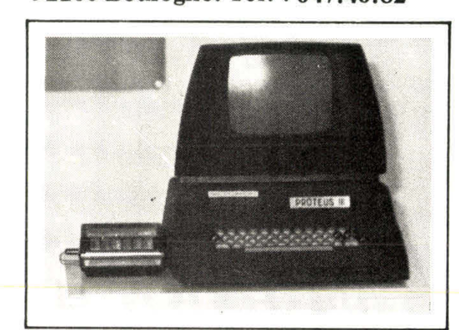
Enfin **Proteus Floppy** est un mini floppy travaillant en double densité, double face. Ses autres caractéristiques : 2,6 Mégabits par disquette, vitesse de transfert 250 K-Bauds, temps d'accès moyen à un fichier 500 ms.

La commercialisation en France a commencé le 15 octobre, le prix de base de l'unité centrale Proteus III est fixé à 7 500 F H.T.

Spécifications techniques de l'unité centrale Proteus III

25 K-de mémoire ou 41 K
16 K-RAM (réf. Proteus III-A)
32 K-RAM (réf. Proteus III-B)
1 K-de mémoire de page écran
8 K-ROM (Basic)
Unité centrale 6800, travaillant à 0,894 MHz
Clavier, 53 touches capacitives
18 caractères dont ASCII majuscules, plus caractères graphiques
Sortie UHF et vidéo
Interface imprimante série : 20 mA, TTY, RS 232, gérée par le basic
BASIC type Etendu, capable de gérer 3 périphériques simultanément (clavier - Ecran vidéo, Imprimante - TTY, Magnéto cassette), plus éventuellement 1 à 3 Floppy disques
Magnétophone à cassettes travaillant à 300 Bauds, utilisé comme mémoire de masse.

PROTEUS INTERNATIONAL S.A., 53, rue de Paris 92100 Boulogne. Tél. : 647.46.82



Le nouveau microcalculateur « une puce » R 6500/1

Le microcalculateur « Une Puce » N-MOS (R 6500/1) fonctionnant à 2 MHz avec un temps d'exécution d'une instruction de 1 microseconde minimum, a été développé par Rockwell International.

Le R 6500/1 à 40 broches a un logiciel compatible avec celui de la famille 6500. Il a un jeu d'instruction identique, y compris les 13 modes d'adressage du CPU 6502.

Il fonctionne à l'aide d'une alimentation unique de 5 volts et se caractérise par une broche d'alimentation séparée permettant à une Mémoire RAM de fonctionner avec 10 % de la puissance.

Sont incorporés à la puce 1 ROM 2k × 8, 1 RAM 64 × 8, 1 intervallomètre/compteur d'impulsion 16 bits et 32 lignes E/S bidirectionnelles.

Il a en outre des interruptions susceptibles d'être masquées ou non-masquées, et une ligne d'entrée impulsion/sortie temporisation.

Les 32 lignes E/S bidirectionnelles sont divisées en quatre broches de huit bits (A, B, C et D).

Rockwell fournit un émulateur à 64 broches parmi lesquelles 40 broches sont électriquement identiques au standard R 6500/1 pour faciliter le développement de systèmes prototypes. Un module de simulation R 6500/1 permet le développement du logiciel et du hardware ; un système d'émulation intégrée est disponible comme option du système de développement microcalculateur de la Société, le « Système 65 ».

Pour toute information complémentaire concernant ces produits.

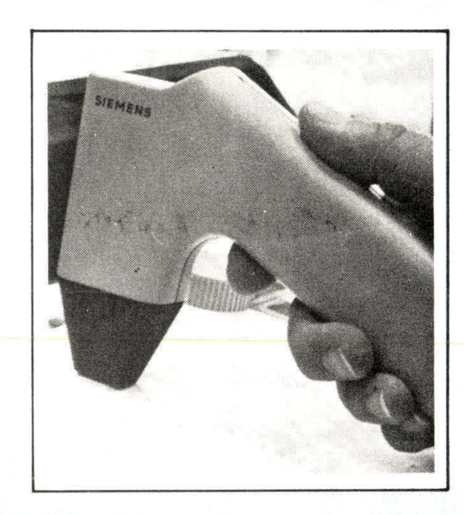
RADIO ÉQUIPEMENTS-ANTARES S.A.
9, Rue Ernest Cognacq
92301 Levallois-Perret

Saisir les données d'un simple geste de la main

Le nouveau lecteur OCR HL5 de Siemens permet de saisir les données à une vitesse dix fois supérieure à celle des procédés classiques. Cet analyseur pourvu d'une poignée et ne

pesant que 150 g est promené d'un simple geste sur la ligne à lire, de gauche à droite, ou de droite à gauche, peu importe. Un signal acoustique indique si tous les caractères ont bien été reconnus et transmis à un terminal à écran ou à cassette.

L'analyseur complet se compose d'un système d'exploitation et d'une unité de détection reliée par un mince câble et équipée de microprocesseurs. L'appareil reconnaît les 32 caractères définis des écritures OCR-A et OCR-B à une vitesse maximale de 140 caractères à la seconde.



Grâce à une interface définie, le lecteur peut transmettre les données à d'autres terminaux, une caisse enregistreuse électronique par exemple. C'est pourquoi cet appareil convient parfaitement aux applications dans le commerce pour la lecture des étiquettes avec indication des références et du prix à la caisse, aux points de vente ainsi que dans l'industrie pour déchiffrer les étiquettes lors de la gestion des stocks, des inventaires ou de la fabrication. Il permet également la lecture des bordereaux de paiement dans les établissements de crédit, des cartes d'affiliation à la sécurité sociale, des actes médicaux et des cartes de toutes sortes établies par l'Administration, pour ne citer que quelques exemples parmi les nombreux débouchés ouverts à cet appareil universel de saisie des données.

Siemens
39, bd Ornano
93 Saint Denis
Tél. : 243-30-20

Hindi Première

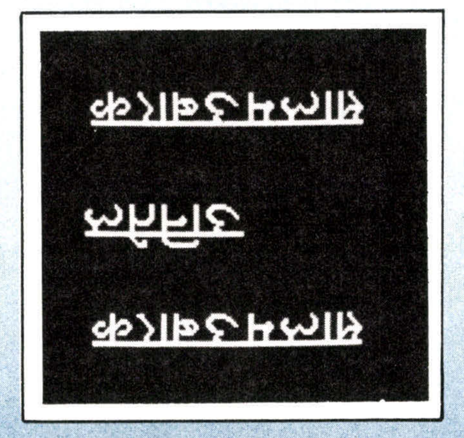
Après l'Arabe, le Grec, le Cyrillique, le Farsi, Unitel est la première société à équiper ses générateurs de caractères en Hindi.

Bien qu'il ne soit pas encore aussi répandu en Inde qu'en Occident, l'audiovisuel s'affirme comme le premier support de l'Hindi, langue officielle, au sein des divers groupements linguistiques.

Dans ses 21 états, 14 langues sont couramment employées. Les Indiens se déplaçant souvent, la nécessité d'un affichage dans les gares, les aéroports s'impose : c'est une première application. La seconde concerne, bien sûr, la télévision : programmes scolaires, culturels et récréatifs, communiqués, traductions, élections...

Vous pouvez voir sur la photographie l'écriture synthétique réalisée par SCRIPTEL « H » : « Bonne année Unitel ».

UNITEL, 80, rue d'Arcueil, 94250 Gentilly – France. Tél. : 588.33.13.



Concours « MICRO »

La Mission à l'Informatique du ministère de l'Industrie organise un **Concours « MICRO »** portant sur l'**utilisation de micro-ordinateurs dans la vie quotidienne.**

Ce concours est ouvert à tous les publics : jeunes ou adultes, informaticiens ou non, se présentant à titre individuel ou en groupe.

L'objet du concours est de proposer un projet ou une réalisation d'utilisation de micro-ordinateur individuel, existant sur le marché, dans des applications de la vie quotidienne (loisir, enseignement, gestion domestique, musique, etc.).

La date de clôture est fixée au 15 mai 1979. Un jury composé de représentants du public et notamment de jeunes et de professionnels de l'informatique, sera constitué par la Mission à l'Informatique et publiera les résultats à partir du 15 septembre 1979.

Les prix seront attribués aux lauréats du concours dans les différentes catégories (jeunes, individuels ou groupes, etc.). Ces prix seront soit en espèces, soit en nature (voyages, matériel) ; la valeur de chacun de ces prix n'excèdera pas 50 000 F. Un titre de « Lauréat du concours MICRO 1979 » sera décerné.

Un dossier complet d'information sur le concours sera disponible à partir de novembre en écrivant à :
Concours « MICRO »
Mission à l'Informatique
24, rue de l'Université
75700 PARIS.

Un nouveau micro-ordinateur : l'AIM de Rockwell

Rockwell représenté en France par R.E.A. introduit sur le marché un micro-ordinateur l'AIM 65 (AIM pour Advance Interface Module).

L'AIM 65 est construit autour du microprocesseur R 6502 et des circuits associés de la famille 6500.

Il rassemble, tous les éléments nécessaires à la mise en œuvre d'un système micro-informatique complet. Se présentant sous la forme de deux cartes de dimensions 25 × 30 cms et 25 × 10, il comprend :

— un véritable clavier de terminal Qwerty à 54 touches permettant de coder 64 caractères ASCII et 11 touches de fonction dont trois programmables par l'utilisateur,

— une visualisation alphanumérique de vingt caractères — 1 ligne complète — par LED 16 segments, et avantage décisif pour une machine de ce prix,

— une imprimante alphanumérique à vingt caractères par ligne, travaillant à la cadence de 90 lignes par minute. 64 caractères peuvent être imprimés sous forme d'une matrice de 5 × 7 points.

Le logiciel système est résident en ROM d'une contenance de 16 K-0 octets dans la version la plus complète.

Les interfaces du système comprennent : une ligne série type télétype, deux interfaces pour un lecteur/enregistreur de cassettes audio (deux formats possibles), deux canaux d'Entrée/Sortie de huit lignes bidirectionnelles, une ligne série et deux lignes d'interface avec les deux timers incorporés au système.

L'AIM 65 est un ordinateur d'usage général qui a été conçu pour :

— le développement et la simulation d'applications avec le 6500,
— l'enseignement de l'informatique,
— l'utilisation par les amateurs souhaitant aller au-delà du stade de l'initiation.

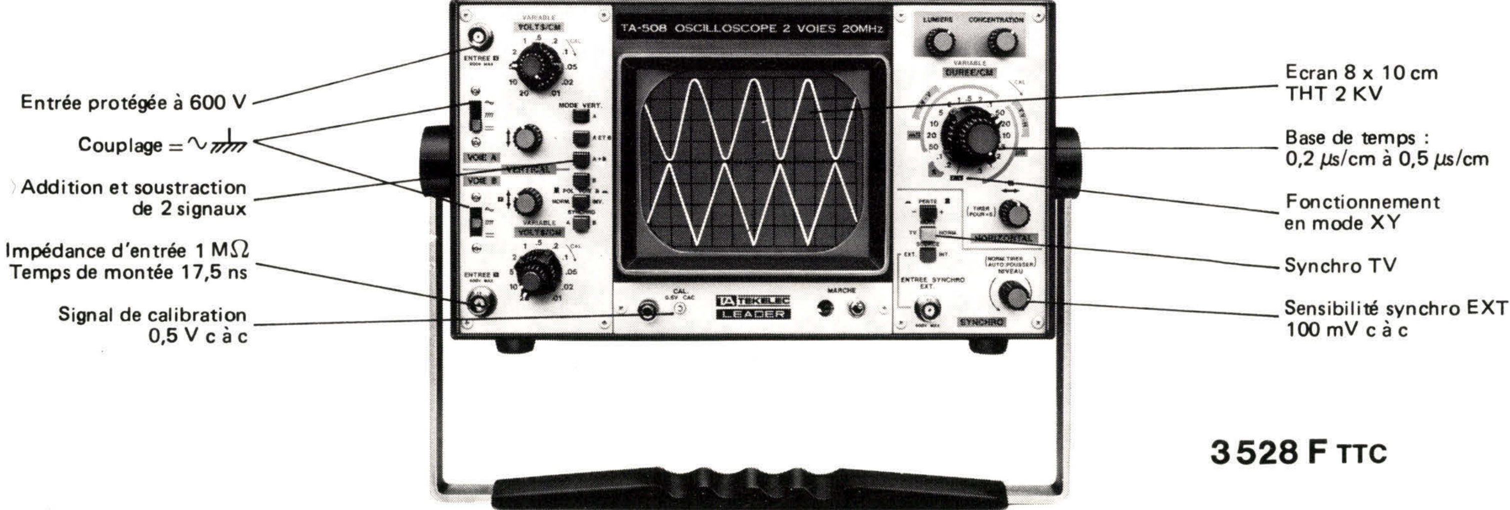
Radio Équipements-Antares
Boîte Postale n° 5
9, rue Ernest-Cognacq
92301 Levallois Perret Cedex
Tél. : 758-11-11

Le directeur de la publication : J.-P. VENTILLARD. – N° de Commission paritaire en cours. – Dépôt légal n° 12898
Photocomposition : Algaprint, 8, rue Monte-Cristo, 75020 Paris, tél. : 370-45-30. – Imprimerie : La Haye-Mureaux, 78130 Les Mureaux, tél. : 261-56-15.

Chez TEKELEC-AIRTRONIC
4 PRODUITS SORTANT DE L'ORDINAIRE

Tarif Juillet 1978

TA/LEADER **TA508** oscilloscope double trace 20 MHz

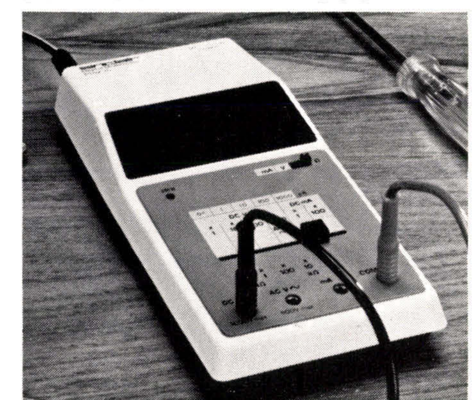


3 528 F TTC

Et toujours la QUALITE LEADER (voir Banc d'Essai HP N° 1631 du mois d'Avril 1978)

SINCLAIR le numérique à la portée de tous

PDM 35 — 395 F TTC



- 1 V à 1000 V = PE
- 1 V à 500 V ∿
- 0,1 µA à 100 mA = PE
- 1 KΩ à 10 MΩ PE

- PILE 9 V
- SECTEUR

Dans la poche
(180 g)

DM 235 *nouveau* 690 F TTC

- PILES 6 V
- BATTERIES
- SECTEUR

- 2000 points : 0,5 %
- 2 V à 1000 V continu PE
- 2 V à 750 V ∿ PE
- 2 mA à 1000 mA = et ∿ PE
- 2 KΩ à 20 MΩ (protection 250 V)

Dans la mallette de dépannage,
le laboratoire
(640 g)

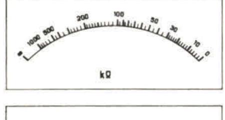


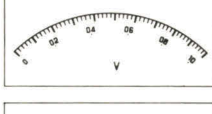
SANWA le premier contrôleur analogique à CRISTAUX LIQUIDES

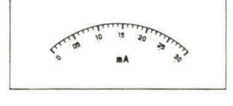
LCD 900

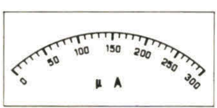


PLUS D'ERREUR DE LECTURE :
UNE SEULE ECHELLE
APPARAIT A LA FOIS









935 F TTC

- 4 gammes en résistance : 1 KΩ à 1000 KΩ
- Précision en tension : ± 3 %
- Dimensions : 200 x 135 x 50 mm
- Poids : 800 g

- 50 KΩ/Volt
- 7 gammes en tension continue : 1 V à 1000 V
- 5 gammes en tension alternative : 10 V à 1000 V (10 KΩ/V)
- 4 gammes en courant continu : ± 0,3 à 300 mA
- 1 gamme en courant alternatif : 3 A

DISTRIBUTEURS PDM 35

PARIS : **ACER**, 42 bis rue de Chabrol (10e) ● **CIBOT**, 13 rue de Reuilly (12e) ● **DCE**, 86 rue de Colombes (Courbevoie) ● **FRANKLIN 2000**, 8 rue de l'Arrivée (15e) ● **CIA**, 63 bd Kellerman (13e) ● **JCS COMPOSANTS**, 35 rue Croix Nivert (15e) ● **PENTASONIC**, 5 rue Maurice Bourdet (16e) ● **Radio MJ**, 17 rue Claude Bernard (5e) ● **RADIO CHAMPERRET**, 12 place Pte Champerret (17e) ● **RADIO PRIM**, 6 allée Verte (11e).

PROVINCE :

OUEST : **ELECTROSOLD**, 21 rue St Martin, Le Mans ● **RADIO PIECES DELAHAYE**, 23 rue de Chateaudun, Rennes ● **RADIO SELL**, 161 av. Jean Jaurès, Brest.

CENTRE : **ECA**, 22 Quai Thanaron, Bourg les Valence ● **MANU MESURE**, 8 av. Mal de Lattre de T., Bron ● **MESURELEC**, 46 Rte de Genas, Chassieu ● **RCB**, rue Ferre - ZI de Crissey, Châlon s/Saône ● **RHONE-ALPES MESURE**, Chemin des Epinettes, Sathonay Village.

NORD : **DECOCK**, 4 rue Colbert, Lille ● **SELECTRONIC**, 14 bd Carnot, Lille.

SUD-OUEST : **COMPTOIR DU LANGUEDOC**, 26 rue du Languedoc, Toulouse ● **COMPTOIR RADIO DU BEARN**, 10 Impasse Henri IV, Gelos ● **SOLISELEC**, 37 Crs d'Alsace-Lorraine, Bordeaux.

SUD : **CITEM**, 31 bd de la Madeleine, Nice ● **POINT ELECTRONIQUE**, 14 rue Roussy, Nîmes.

EST : **FACHOT**, Metz ● **COMELEC**, 66 rue de Metz, Longwy ● **HBN**, 12 rue Gauchette, Reims.

DISTRIBUTEURS LEADER

PARIS : **ACER**, 42 bis rue de Chabrol (10e) ● **DCE**, 86 rue de Colombes (Courbevoie) ● **FANATRONIC**, 35 rue Croix Nivert (15e) ● **PENTASONIC**, 5 rue Maurice Bourdet (16e).

PROVINCE : **CITEM**, 31 bd de la Madeleine, Nice ● **COMPTOIR DU LANGUEDOC** 26 rue du Languedoc, Toulouse ● **DECOCK**, 4 rue Colbert, Lille ● **SOLISELEC**, 37 Crs d'Alsace-Lorraine, Bordeaux.

760 TP

DIVISION COMMERCIALE INSTRUMENTS - BP N° 2 - 92310 SEVRES - Tél. : (1) 534-75-35 - Télex : TEKLEC 204 552 F

TEKELEC TA AIRTRONIC



FAIRCHILD